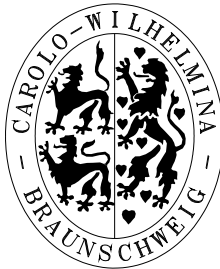


Classification and Overview of Meshfree Methods

Thomas-Peter Fries, Hermann-Georg Matthies
Institute of Scientific Computing
Technical University Braunschweig
Brunswick, Germany

Informatikbericht Nr.: 2003-3

July, 2004
(revised)



Classification and Overview of Meshfree Methods

Thomas-Peter Fries, Hermann-Georg Matthies
Department of Mathematics and Computer Science
Technical University Braunschweig
Brunswick, Germany

July, 2004
(revised)

Location

Institute of Scientific Computing
Technical University Braunschweig
Hans-Sommer-Strasse 65
D-38106 Braunschweig

Postal Address

Institut für Wissenschaftliches Rechnen
Technische Universität Braunschweig
D-38092 Braunschweig
Germany

Contact

Phone: +49-(0)531-391-3000
Fax: +49-(0)531-391-3003
E-Mail: wire@tu-bs.de
www: <http://www.tu-bs.de/institute/WiR>

Copyright © by Institut für Wissenschaftliches Rechnen, Technische Universität Braunschweig

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted in connection with reviews or scholarly analysis. Permission for use must always be obtained from the copyright holder.

Alle Rechte vorbehalten, auch das des auszugsweisen Nachdrucks, der auszugsweisen oder vollständigen Wiedergabe (Photographie, Mikroskopie), der Speicherung in Datenverarbeitungsanlagen und das der Übersetzung.

Classification and Overview of Meshfree Methods

Thomas-Peter Fries, Hermann-Georg Matthies
July, 2004
(revised)

Abstract

This paper gives an overview of Meshfree Methods. Starting point is a classification of Meshfree Methods due to three aspects: The construction of a partition of unity, the choice of an approximation either with or without using an extrinsic basis and the choice of test functions, resulting into a collocation, Bubnov-Galerkin or Petrov-Galerkin Meshfree Method. Most of the relevant Meshfree Methods are described taking into account their different origins and viewpoints as well as their advantages and disadvantages. Typical problems arising in meshfree methods like integration, treatment of essential boundary conditions, coupling with mesh-based methods etc. are discussed. Some valuing comments about the most important aspects can be found at the end of each section.

Contents

1	Introduction	4
2	Preliminaries	9
2.1	Nomenclature	9
2.2	Abbreviations	9
2.3	Method of Weighted Residuals	11
2.4	Complete Basis	12
2.5	Consistency and Partition of Unity (PU)	13
3	Classification of Meshfree Methods	14
4	Construction of a Partition of Unity	15
4.1	Mesh-based Construction	17
4.2	Moving Least-Squares (MLS)	18
4.2.1	Deduction by Minimization of a Weighted Least-Squares Functional	18
4.2.2	Deduction by Taylor Series Expansion	22
4.2.3	Deduction by Direct Imposition of the Consistency Conditions	26
4.2.4	Generalized Moving Least-Squares (GMLS)	28
4.2.5	Relation to Shepard's Method	28
4.2.6	Relation to other Least-Squares Schemes	30
4.3	Reproducing Kernel Particle Method (RKPM)	32
4.3.1	Hermite RKPM	40
4.4	Particle Placement	41
4.5	Weighting Functions	41
4.6	Solving the $k \times k$ System of Equations	47
4.7	Summary and Comments	49

5	Specific Meshfree Methods	51
5.1	Smoothed Particle Hydrodynamics (SPH, CSPH, MLSPH) . . .	52
5.2	Diffuse Element Method (DEM)	59
5.3	Element Free Galerkin (EFG)	61
5.4	Least-squares Meshfree Method (LSMM)	63
5.5	Meshfree Local Petrov-Galerkin (MLPG)	63
5.6	Local Boundary Integral Equation (LBIE)	66
5.7	Partition of Unity Methods (PUM, PUFEM, GFEM, XFEM) . .	68
5.8	<i>hp</i> -clouds	71
5.9	Natural Element Method (NEM)	73
5.10	Meshless Finite Element Method (MFEM)	74
5.11	Reproducing Kernel Element Method (RKEM)	75
5.12	Others	76
5.13	Summary and Comments	77
6	Related Problems	79
6.1	Essential Boundary Conditions	79
6.1.1	Lagrangian Multipliers	79
6.1.2	Physical Counterpart of Lagrangian Multipliers	80
6.1.3	Penalty Approach	80
6.1.4	Nitsche's Method	81
6.1.5	Coupling with Finite Elements	81
6.1.6	Transformation Method	82
6.1.7	Singular Weighting Functions	83
6.1.8	PUM Ideas	84
6.1.9	Boundary Collocation	84
6.1.10	D'Alembert's Principle	85
6.2	Integration	86
6.2.1	Direct Nodal Integration	89
6.2.2	Integration with Background Mesh or Cell Structure . . .	89

6.2.3	Integration over Supports or Intersections of Supports . .	91
6.3	Coupling Meshfree and Mesh-based Methods	91
6.3.1	Coupling with a Ramp Function	93
6.3.2	Coupling with Reproducing Conditions	95
6.3.3	Bridging Scale Method	98
6.3.4	Coupling with Lagrangian Multipliers	100
6.4	Discontinuities	100
6.4.1	Visibility Criterion	101
6.4.2	Diffraction Method	102
6.4.3	Transparency Method	103
6.4.4	PUM Ideas	103
6.5	<i>h</i> -Adaptivity	103
6.6	Parallelization	106
6.7	Solving the Global System of Equations	107
6.8	Summary and Conclusion	108
7	Conclusion	109
	References	111

1 Introduction

Numerical methods are indispensable for the successful simulation of physical problems as the underlying partial differential equations usually have to be approximated. A number of methods have been developed to accomplish this task. Most of them introduce a finite number of nodes and can be based on the principles of weighted residual methods.

Conventional numerical methods need an *a priori* definition of the connectivity of the nodes, i.e. they rely on a mesh. The Finite Element Method (FEM) and Finite Volume Method (FVM) may be the most well-known members of these thoroughly developed mesh-based methods. In contrast, a comparably new class of numerical methods has been developed which approximates partial differential equations only based on a set of nodes without the need for an additional mesh. This paper is devoted to these *Meshfree Methods* (MMs).

A multitude of different MMs has been published during the last three decades. Despite the variety of names of individual methods it is interesting to note that in fact there are significant similarities between many of these methods. Here, we try to put most of the MMs into a unified context and discuss a classification of MMs. However, it is also our aim not to neglect individual aspects of the particular methods, mentioning their different origins and viewpoints. Other surveys on MMs may for example be found in [17] by Belytschko *et al.* and in [87] by Li and Liu. The first focuses also on the similarities between the different meshfree methodologies, whereas the latter gives an extensive listing of applications of MMs in practice. Special issues of journals on various aspects of MMs may be found in [27, 28, 93]. A few books on MMs are also available, see e.g. [4, 56, 91].

Features of Meshfree Methods Before giving an outline of the paper we list some of the most important features of MMs, often comparing them with the analogous properties of mesh-based methods:

- Absence of a mesh
 - In MMs the connectivity of the nodes is determined at run-time.
 - No mesh alignment sensitivity. This is a serious problem in mesh-based calculations e.g. of cracks and shear bands [87].
 - h -adaptivity is comparably simple with MMs as only nodes have to be added, and the connectivity is then computed at run-time automatically. p -adaptivity is also conceptionally simpler than in mesh-based methods.

- No mesh generation at the beginning of the calculation is necessary. This is still not a fully automatic process, especially not in complex three-dimensional domains, and may require major human interactions [69].
- No remeshing during the calculation. Especially in problems with large deformations of the domain or moving discontinuities a frequent remeshing is needed in mesh-based methods, however, a conforming mesh with sufficient quality may be impossible to maintain. Even if it is possible, the remeshing process degrades the accuracy considerably due to the perpetual projection between the meshes [17], and the post-processing in terms of visualization and time-histories of selected points requires a large effort [19].
- Continuity of shape functions: The shape functions of MMs may easily be constructed to have any desired order of continuity.
 - MMs readily fulfill the requirement on the continuity arising from the order of the problem under consideration. In contrast, in mesh-based methods the construction of even C^1 continuous shape functions — needed e.g. for the solution of fourth order boundary value problems — may pose a serious problem [88].
 - No post-processing is required in order to determine smooth derivatives of the unknown functions, e.g. smooth strains.
 - Special cases where the continuity of the meshfree shape functions and derivatives is not desirable, e.g. in cases where physically justified discontinuities like cracks, different material properties etc. exist, can be handled with certain techniques.
- Convergence: For the same order of consistency numerical experiments suggest that the convergence results of the MMs are often considerably better than the results obtained by mesh-based shape functions [85]. However, theory fails to predict this higher order of convergence [85].
- Computational effort: In practice, for a given reasonable accuracy, MMs are often considerably more time-consuming than their mesh-based counterparts.
 - Meshfree shape functions are of a more complex nature than the polynomial-like shape functions of mesh-based methods. Consequently,

the number of integration points for a sufficiently accurate evaluation of the integrals of the weak form is considerably larger in MMs than in mesh-based methods. In collocation MMs no integration is required, however, this advantage is often compensated by evoking accuracy and stability problems.

- At each integration point the following steps are often necessary to evaluate meshfree shape functions: Neighbour search, solution of small systems of equations and small matrix-matrix and matrix-vector operations in order to determine the derivatives.
- The resulting global system of equations has in general a larger bandwidth for MMs than for comparable mesh-based methods [17].
- Essential boundary conditions: Most MMs lack Kronecker delta property, i.e. the meshfree shape functions Φ_i do not fulfill $\Phi_i(\mathbf{x}_j) = \delta_{ij}$. This is in contrast to mesh-based methods which often have this property. Consequently, the imposition of essential boundary conditions requires certain attention in MMs and may degrade the convergence of the method [60].
- Locking: Differently from what has been stated in early papers [18] it should be mentioned that MMs may as well suffer from the locking phenomenon, similarly to the FEM, see [31, 67]. It is sometimes possible to alleviate this phenomenon by tuning some parameters of the MM.

Outline of the Paper The references given in the following outline are restricted to only a few important publications; later on, in the individual subsections, a number of more references are mentioned. The paper is organized as follows: Section 2 aims to introduce abbreviations and some important mathematical terms which then will be used frequently in the rest of the paper.

In section 3 we propose a classification of MMs. According to this classification the MMs fall clearly into certain categories and their differences and relations can be seen. We do not want to overemphasize the *meshfree* aspect—although being the main issue of this paper—, because some methods can also employ mesh-based interpolations e.g. from the FEM. We classify MMs according to

- the construction of a partition of unity of n -th order with an intrinsic basis
- the choice of the approximation which can use an intrinsic basis only or add an extrinsic basis

- the choice of the test function in the weighted residual procedure which might lead to collocation procedures, Bubnov-Galerkin methods etc.

Section 4 describes the meshfree construction of a partition of unity (PU) which is the starting point for MMs. The Moving Least-Squares (MLS) [82] procedure or the Reproducing Kernel Particle Method (RKPM) [94, 95, 98] is most often used for this purpose. Different ways are shown to obtain the MLS functions and relations to other least-square schemes and Shepard functions are pointed out. Then, the RKPM is deduced using different ways. It can finally be seen that although the two principles MLS and RKPM have their roots in very different areas—the first has its origin in data fitting, the second in wavelet theory—the resulting partition of unity functions are almost (in practice often exactly) the same.

In section 5 most of the relevant MMs are discussed in more detail. Although the most important characteristics of each method can already be seen from the classification in section 3, in this section many more details are given, considering the different viewpoints and origins of each method. Also problems and advantages of each method are discussed.

We start with MMs based on approximations that use the concept of an intrinsic basis only. The collocation MMs, like the Smoothed Particle Hydrodynamics (SPH) [104, 108] as the earliest MM and some of its important variations (Corrected SPH [22] and Moving Least-Squares SPH [38]) are described in section 5.1. Similar methods like collocation RKPM [1] and the Finite Point Method (FPM) [114, 115] are also briefly mentioned here. Then, the Diffuse Element Method (DEM) [111] and the Element Free Galerkin (EFG) method [18] as members of Bubnov-Galerkin MMs are described in 5.2 and 5.3. While the first suffers from a number of problems due to some simplifications, the latter may be considered the “fixed” version, being a very popular MM in today’s practice. The Meshless Local Petrov-Galerkin (MLPG) method [4] is the topic of subsection 5.5, it falls into a number of different versions, MLPG 1 to MLPG 6, which depends on the choice of the test functions. The term *local* in the name MLPG refers to the fact that local weak forms of partial differential equations are employed which differs slightly from standard methods based on global weak forms. Subsection 5.6 discusses the Local Boundary Integral Equation (LBIE) [5, 133], which is the meshfree version of conventional boundary element methods. It can also be considered to be a certain version of the MLPG methods.

Then, MMs based on the usage of an additional extrinsic basis are described. These methods allow to increase the order of consistency of an existing partition

of unity and/or to include a certain knowledge of the solution's behaviour into the approximation space. The resulting global system of equations grows in certain factors depending on the size of the extrinsic basis. Typical members of this class are the Partition of Unity Method (PUM) [11], the Partition of Unity Finite Element Method (PUFEM) [106], the Generalized Finite Element Method (GFEM) [123, 124], the Extended Finite Element Method (XFEM) [19] and the *hp*-cloud method [42, 43], discussed in subsections 5.7 - 5.8.

Sections 5.9 - 5.11 describe some “non-standard” approaches of MMs. The Natural Element Method (NEM) [125, 126] uses Sibson or non-Sibsonian interpolations, which are constructed in terms of Voronoi cells; the Meshless Finite Element Method (MFEM) [69] follows this idea closely. The Reproducing Kernel *Element* Method (RKEM) [96] may be used to construct arbitrary smooth element shape functions overcoming one of the most serious problems of the FEM. One might argue whether or not these special methods are really mesh-free; they somehow combine certain meshfree and mesh-based advantages.

Section 6 is about problems which frequently occur when dealing with mesh-free methods. Subsection 6.1 shows the treatment of essential boundary conditions (EBCs). Due to the lack of Kronecker-delta property of the meshfree shape functions it is not trivial (as in FEM) to impose EBCs. A number of different ideas has been developed [29, 46, 75, 128]. The most frequently used are Lagrangian multipliers, penalty approaches, coupling with finite elements, transformation methods and boundary collocation.

Another issue is the integration in MMs, discussed in 6.2. The meshfree shape functions and their derivatives are rational functions which show a more and more non-polynomial character with rising order of derivatives. This makes the evaluation of the integrals in the weak form of the partial differential equation difficult, requiring large numbers of integration points. There are different concepts for the integration [12, 30, 39]: Background meshes, cell structures, local integration over supports of test functions or over intersections of supports.

Coupling of meshfree and mesh-based methods [20, 65, 129] is an important topic to combine the advantages of both concepts and is discussed in subsection 6.3. Subsection 6.4 is about the treatment of discontinuities such as cracks and material interfaces [16, 21, 116]. The visibility criterion, diffraction method and transparency methods are explained as well as PUM ideas for the enrichment around the discontinuity. *h*-adaptivity in MMs [50, 101, 105] is the topic of subsection 6.5. Finally, some aspects of parallelization of MMs [36, 55] and solving the final system of equations constructed by MMs [54, 83] are mentioned briefly in subsections 6.6 and 6.7.

In section 7 we conclude this paper with a short summary.

2 Preliminaries

2.1 Nomenclature

Throughout this paper we use normal Latin or Greek letters for functions and scalars. Bold small letters are in general used for vectors and bold capital letters for matrices. The following table gives a list of all frequently used variables and their meaning.

symbol	meaning
u	function
u^h	approximated function
\mathbf{x}	space coordinate
\mathbf{x}_i	position of a node (=particle, point)
Φ	shape (=trial, ansatz) functions
Ψ	test functions
\mathbf{N}	FEM shape function (if difference is important)
w	weighting (=window, kernel) function
\mathbf{p}	intrinsic or extrinsic basis
\mathbf{a}	vector of unknown coefficients
\mathbf{M}	moment matrix
α	multi-index
α^i	vector in the multi-index set $\{\alpha \mid \alpha \leq c, c \in \mathbb{N}\}$
d	dimension
n	order of consistency
k	size of a complete basis
N	total number of nodes (=particles, points)
ρ	dilatation parameter (=smoothing length)

2.2 Abbreviations

APET:	Amplitude and Phase Error Term
BC:	Boundary Condition
BEM:	Boundary Element Method
BIE:	Boundary Integral Equation
CSPH:	Corrected Smoothed Particle Hydrodynamics

DEM:	Diffuse Element Method
DOI:	Domain Of Influence
EFG:	Element Free Galerkin
EBC:	Essential Boundary Condition
FEM:	Finite Element Method
FDM:	Finite Difference Method
FLS:	Fixed Least-Squares
FVM:	Finite Volume Method
FPM:	Finite Point Method
GFEM:	Generalized Finite Element Method
GMLS:	Generalized Moving Least-Squares
LBIE:	Local Boundary Integral Equation
LSQ:	Standard Least-Squares
MFEM:	Meshless Finite Element Method
MFLS:	Moving Fixed Least-Squares
MFS:	Method of Finite Spheres
MLPG:	Meshless Local Petrov-Galerkin
MLS:	Moving Least-Squares
MLSPH:	Moving Least-Squares Particle Hydrodynamics
MLSRK:	Moving Least-Squares Reproducing Kernel
MM:	Meshfree Method
NEM:	Natural Element Method
PDE:	Partial Differential Equation
PN:	Partition of Nullity

PU:	Partition of Unity
PUM:	Partition of Unity Method
PUFEM:	Partition of Unity Finite Element Method
RKM:	Reproducing Kernel Method
RKEM:	Reproducing Kernel Element Method
RKPM:	Reproducing Kernel Particle Method
SPH:	Smoothed Particle Hydrodynamics

2.3 Method of Weighted Residuals

The aim is to solve partial differential equations (PDEs) numerically, i.e. we are interested in finding the functions u that fulfill the PDE $\mathcal{L}u = f$ where \mathcal{L} is any differential operator and f is the system's right hand side.

One of the most general techniques for doing this is the weighted residual method. Conventional methods like the Finite Element Method (FEM) are the most popular mesh-based representatives of this method and also the Finite Difference Method (FDM) and the Finite Volume Method (FVM) can be deduced with help of the weighted residual method as the starting point. All Mesh-free Methods (MMs) can also be seen as certain realizations of the weighted least-squares idea.

In this method an approximation of the unknown field variables u is made in summation expressions of trial functions Φ (also called shape or ansatz functions) and unknown nodal parameters $\hat{\mathbf{u}}$, hence $u \approx u^h = \Phi^T \hat{\mathbf{u}} = \sum_i \Phi_i \hat{u}_i$. Replacing u with u^h in the PDE gives $\mathcal{L}u^h - f = \varepsilon$. As it is in general not possible to fulfill the original PDE exactly with the approximation a residual error ε is introduced. Test functions Ψ are chosen and the system of equations is determined by setting the residual error ε orthogonal to this set of test functions, $\int \Psi \varepsilon d\Omega = \int \Psi (\mathcal{L}u^h - f) d\Omega = 0$. The integral expressions of this weak form of the PDE have to be evaluated with respect to Φ and Ψ , and the given boundary conditions have to be considered. The resulting system of equations $\mathbf{A}\hat{\mathbf{u}} = \mathbf{b}$ is to be solved for determining the unknowns $\hat{\mathbf{u}}$. Throughout this paper we often write \mathbf{u} instead of $\hat{\mathbf{u}}$.

It should be mentioned that one makes often use of the divergence theorem during this procedure to modify the integral expressions in order to shift conditions between the trial and test functions.

In other contexts the test functions Ψ are sometimes also termed weighting functions. But in the context of MMs one should strictly distinguish between test and weighting functions because the term weighting function is already used in a different context of MMs.

2.4 Complete Basis

The concept of a complete basis is of high importance in the framework of MMs, in particular the complete *polynomial* basis up to a certain order. A convenient way to formulate a complete basis is possible with help of the multi-index notation. The multi-index $\alpha = (\alpha_1, \dots, \alpha_d)$ is used in the following with $\alpha_i \geq 0 \in \mathbb{N}$ and $d = \dim(\Omega)$ being the dimension of the problem. If α is applied to a vector \mathbf{x} of the same length then

$$\mathbf{x}^\alpha = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_d^{\alpha_d},$$

and $D^\alpha u(\mathbf{x})$ is the Fréchet derivative of the function u , that is

$$D^\alpha u(\mathbf{x}) = \frac{\partial^{|\alpha|} u(\mathbf{x})}{\partial^{\alpha_1} x_1 \partial^{\alpha_2} x_2 \cdots \partial^{\alpha_d} x_d}.$$

The length of α is $|\alpha| = \sum_{i=1}^d \alpha_i$.

With this notation we can easily define a polynomial basis of order n as

$$\mathbf{p}(\mathbf{x}) = \{\mathbf{x}^\alpha \mid |\alpha| \leq n\}. \quad (2.1)$$

Some examples of complete bases in one and two dimensions ($d = 1, 2$) for first and second order consistency ($n = 1, 2$) are

$$\begin{aligned} \begin{array}{l} d=1 \\ n=1 \end{array} : \quad \{\alpha : |\alpha| \leq 1\} &= \left\{ \begin{array}{c} (0) \\ (1) \end{array} \right\} &\Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0)} \\ \mathbf{x}^{(1)} \end{bmatrix} &= \begin{bmatrix} 1 \\ x \end{bmatrix}, \\ \begin{array}{l} d=1 \\ n=2 \end{array} : \quad \{\alpha : |\alpha| \leq 2\} &= \left\{ \begin{array}{c} (0) \\ (1) \\ (2) \end{array} \right\} &\Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0)} \\ \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{bmatrix} &= \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}, \\ \begin{array}{l} d=2 \\ n=1 \end{array} : \quad \{\alpha : |\alpha| \leq 1\} &= \left\{ \begin{array}{c} (0,0) \\ (1,0) \\ (0,1) \end{array} \right\} &\Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0,0)} \\ \mathbf{x}^{(1,0)} \\ \mathbf{x}^{(0,1)} \end{bmatrix} &= \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}, \\ \begin{array}{l} d=2 \\ n=2 \end{array} : \quad \{\alpha : |\alpha| \leq 2\} &= \left\{ \begin{array}{c} (0,0) \\ (1,0) \\ (0,1) \\ (2,0) \\ (1,1) \\ (0,2) \end{array} \right\} &\Rightarrow \mathbf{p} = \begin{bmatrix} \mathbf{x}^{(0,0)} \\ \mathbf{x}^{(1,0)} \\ \mathbf{x}^{(0,1)} \\ \mathbf{x}^{(2,0)} \\ \mathbf{x}^{(1,1)} \\ \mathbf{x}^{(0,2)} \end{bmatrix} &= \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix}. \end{aligned}$$

It can be seen that although α is a vector, the set of all vectors α with $|\alpha| \leq n$, hence $\{\alpha : |\alpha| \leq n\}$ can be considered a matrix. In the following α^i refers to the α -vector in the i -th line of the set $\{\alpha : |\alpha| \leq n\}$, whereas α_j stands for a specific component of a certain vector α^i .

The relationship between the dimension d and the consistency n on the one hand and the number of components in the basis vector on the other hand is

$$k = \frac{1}{d!} \prod_{i=1}^d (n+i).$$

Consequently, in one dimension we have $k = (n+1)$, in two dimensions $k = 1/2(n+1)(n+2)$ and in three dimensions $k = 1/6(n+1)(n+2)(n+3)$.

2.5 Consistency and Partition of Unity (PU)

In a mathematical sense a scheme $\mathcal{L}_h u = f$ is consistent of order p with the differential equation $\mathcal{L}u = f$ if $\|\mathcal{L}u - \mathcal{L}_h u\| = O(h^n)$, where h is some measure of the node density. It is obvious that the approximation error $\|\mathcal{L}u - \mathcal{L}_h u\|$ goes to zero if $h \rightarrow 0$. There is a relation of the convergence rate and consistency (assuming stability is fulfilled).

Consistency also refers to the highest polynomial order which can be represented *exactly* with the numerical method. In this paper, the term consistency is used in this sense. If the trial functions in a method of weighted residuals have n -th order consistency then the analytical solution up to this order can be found exactly. If the analytical solution is of higher order than n then an approximation error occurs. Depending on the order of a PDE there are certain consistency requirements. For example, approximating a PDE of order $2l$ with a Galerkin method (where the weak form is considered) requires test and shape functions with l -th order consistency.

The terms completeness and reproducing ability are very closely related to consistency [15]. It may be claimed that completeness is closer related to the analysis of Galerkin methods, whereas consistency is more related to collocation techniques [15], however this difference is not further relevant throughout this paper.

A set of functions $\{\Phi_i\}$ is consistent of order n if the following consistency (=reproducing) conditions are satisfied

$$\sum_i \Phi_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i) = \mathbf{p}(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega, \quad (2.2)$$

where \mathbf{p} is the complete basis of Eq. 2.1. The derivative reproducing conditions follow immediately as

$$\sum_i D^\alpha \Phi_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i) = D^\alpha \mathbf{p}(\mathbf{x}). \quad (2.3)$$

The set $\{\Phi_i\}$ is also called *partition of unity* (PU) of order n . The PU of order 0 fulfills $\sum_i \Phi_i(\mathbf{x}) = 1$ —as $p_1(\mathbf{x}) = \mathbf{x}^{\alpha_1} = \mathbf{x}^{(0,\dots,0)} = 1$ according to Eq. 2.1—which reflects the basic meaning of the term partition of unity. In a weighted residual method having an approximation of the form $u \approx u^h = \Phi^T \hat{\mathbf{u}} = \sum_i \Phi_i \hat{u}_i$ where $\{\Phi_i\}$ builds a PU of order n it is possible to find a polynomial solution of a PDE under consideration up to this order *exactly*.

Another way to show that the functions Φ_i are n -th order consistent is to insert the terms of the approximation into the Taylor series and identify the resulting error term, i.e. the term in the series which cannot be captured with the approximation. This will be worked out later. In multi-index notation the Taylor series may be written as

$$u(\mathbf{x}_i) = \sum_{|\alpha|=0}^{\infty} \frac{(\mathbf{x}_i - \mathbf{x})^\alpha}{|\alpha|!} D^\alpha u(\mathbf{x}).$$

The construction of meshfree (and mesh-based) functions that build a PU of a certain order is discussed in detail in section 4.

3 Classification of Meshfree Methods

We present the classification of MMs already at this point of the paper—rather than at the end—because it shall serve the reader as a guideline throughout this paper; the aim is not to get lost by the large number of methods and aspects in the meshfree numerical world. Therefore, we hope this to be advantageous for newcomers to this area as well as for thus with a certain prior knowledge.

In this paper, we do not only restrict ourselves to the meshfree aspect—although being the major concern—, as some methods can use either mesh-based or meshfree PUs or even a combination of both via coupling. Therefore we focus the overview in Fig. 1 on the PU of n -th order. The PU can be constructed either with a mesh-based FEM procedure or the meshfree MLS or RKPM principle; other possibilities are also mentioned in the figure and are discussed later. These

techniques for the construction of a PU of n -th order with the concept of the complete (intrinsic) basis $\mathbf{p}(\mathbf{x})$ will be worked out in section 4.

On the basis of the PU the approximation is chosen. If one simply defines $u^h = \sum_i \Phi_i \hat{u}_i$, i.e. if the PU functions are directly taken to be the shape functions, we only use the intrinsic basis needed for the construction of the PU functions. We can also define different approximations using the concept of an *extrinsic* basis $\mathbf{p}(\mathbf{x})$ which may be used either to increase the order of consistency of the approximation, or to include *a priori* knowledge of the exact solution of the PDE into the approximation.

The choice of a test function forms the last step in the characterization of MMs. For example using $\Psi_i = \delta(\mathbf{x}_i - \mathbf{x})$, the Dirac delta function, a collocation scheme will result, or using $\Psi_i = \Phi_i$, a Bubnov-Galerkin procedure follows.

Once more we summarize these three classifying steps of a MM:

- Construction of a PU of n -th order with an intrinsic basis
- Choice of an approximation which can use the intrinsic basis only or add an extrinsic basis
- Choice of the test functions

For a specific MM, these three properties are in general defined. Few methods may occur in more than one case, e.g. the PUM can be used in a collocation or Galerkin scheme. All the specific MMs resulting as certain realizations of the three classifying aspects will be discussed in detail in section 5.

The grey part in Fig. 1 refers to alternative non-standard approaches to construct PUs. Here, Sibson and non-Sibsonian interpolations being the starting point for the NEM (subsection 5.9) are mentioned for example. Also construction ideas which are a combination of meshfree and mesh-based approaches such as thus resulting from coupling methods (subsection 6.3), MFEM functions (subsection 5.10) and RKEM functions (subsection 5.11) belong to these alternative approaches. Calling these alternative approaches meshfree is not without a certain conflict, in fact, they are strongly influenced by both meshfree and mesh-based methods, and try to combine advantages of both methodologies.

4 Construction of a Partition of Unity

In this section several ways are shown for the construction of a PU of order n . For a certain completeness and comparison purposes we start by reviewing the

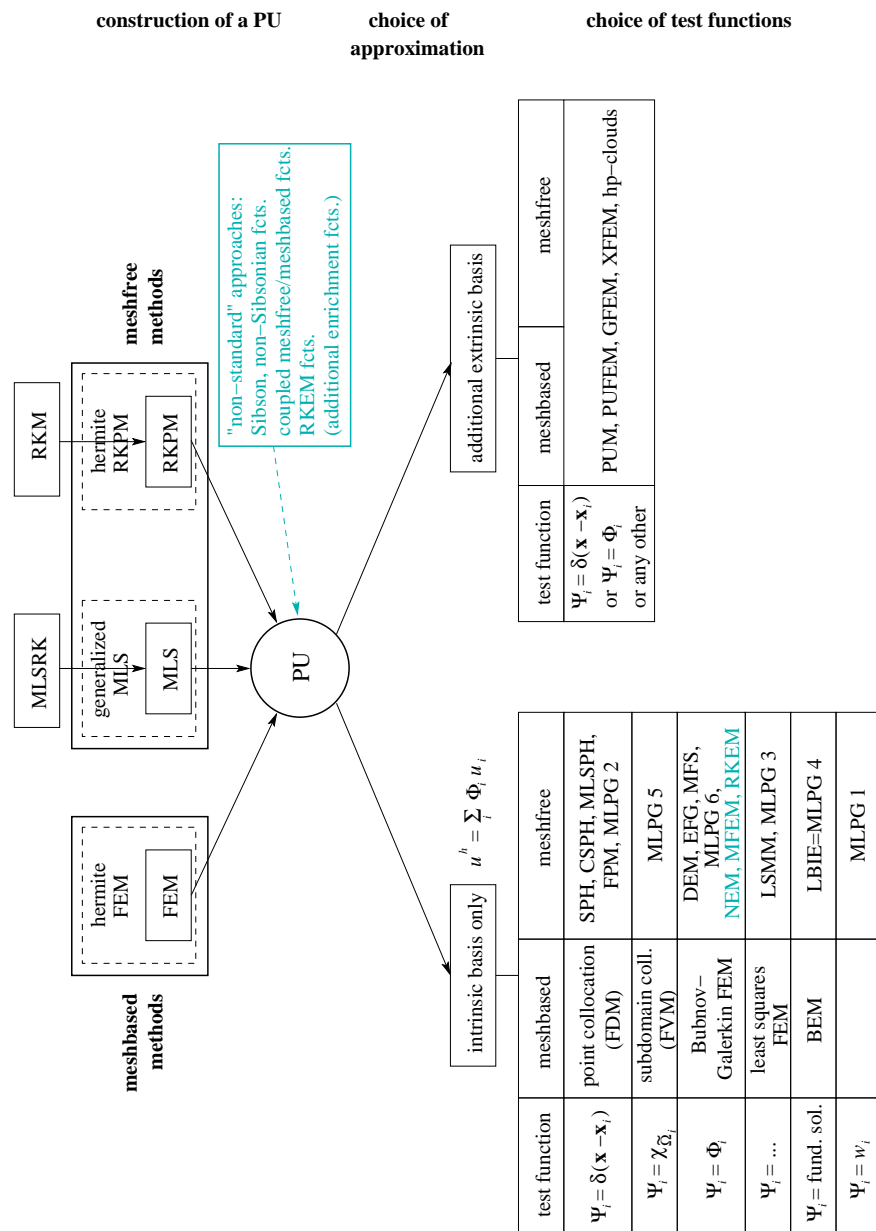


Figure 1: Classification of Meshfree Methods.

construction based on a mesh which may be used as a technique to construct standard linear, quadratic, \dots FEM shape functions. Then, the MLS procedure is deduced in several ways, by minimization of a weighted error functional, by a Taylor series expansion and by direct imposition of the consistency conditions. The RKPM technique —having a very different starting point— turns out to be an equivalent procedure with (almost) the same result than the MLS concept; the same ways for deducing the PU than for the MLS can be chosen here. At the end of this section important aspects of the MLS/RKPM idea are worked out in further detail. The aspects of a suitable node distribution, weighting functions and solution of the $k \times k$ systems of equations which arise in the MLS/RKPM procedures are worked out there.

There are a few MMs that use different ways for the construction of a PU, namely for example the RKEM and NEM. Also coupling methods which combine meshfree and mesh-based ideas to construct a coupled PU may be considered here. These ideas are discussed later on either in section 5 or 6.3. In this section the focus is on the MLS and RKPM procedure which is the central aspect of most MMs.

4.1 Mesh-based Construction

Constructing a PU of n -th order consistency based on a mesh leads to the well known shape functions which are e.g. used in the FEM. The domain Ω is subdivided into non-overlapping finite elements thus leading to a mesh. The PU functions are in general polynomials having the so-called Kronecker delta property meaning that $\Phi_i(\mathbf{x}_j) = \delta(\mathbf{x}_i - \mathbf{x}_j) = \delta_{ij}$.

We want to express the PU functions $\Phi_i(\mathbf{x})$ as polynomials, hence $\Phi_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}_i$, where $\mathbf{p}(\mathbf{x})$ is a complete polynomial basis as explained in subsection 2.4 and \mathbf{a}_i are unknown coefficient vectors being different for each function $\Phi_i(\mathbf{x})$. In index notation we can write $\Phi_i(\mathbf{x}) = p_k(\mathbf{x}) a_{ik}$. To obtain the unknown coefficients a_{ik} we impose Kronecker delta property at the n^e node positions \mathbf{x}_j of the element, consequently $\Phi_i(\mathbf{x}_j) = \delta_{ij}$. This leads to

$$\begin{aligned}\Phi_i(\mathbf{x}_j) &= p_k(\mathbf{x}_j) a_{ik} = \delta_{ij} \\ &\Rightarrow \begin{pmatrix} \mathbf{p}^T(\mathbf{x}_1) \\ \vdots \\ \mathbf{p}^T(\mathbf{x}_{n^e}) \end{pmatrix} \mathbf{A} = \mathbf{I} \\ &\Rightarrow \mathbf{A} = [\mathbf{P}(\mathbf{x}_j)]^{-1} \\ \Phi^T(\mathbf{x}) &= \mathbf{p}^T(\mathbf{x}) [\mathbf{P}(\mathbf{x}_j)]^{-1}.\end{aligned}$$

The conditions for a PU do not have to be imposed directly, but will be satisfied automatically for all functions Φ_i . All shape functions of the FEM build PUs of a certain order (except some special cases, e.g. of p -enriched PUs with bubble functions).

4.2 Moving Least-Squares (MLS)

4.2.1 Deduction by Minimization of a Weighted Least-Squares Functional

The MLS was introduced by Lancaster and Salkauskas in [82] for smoothing and interpolating data. If a function $u(x)$ defined on a domain $\Omega \in \mathbb{R}^d$ is sufficiently smooth, one can define a “local” approximation around a fixed point $\bar{\mathbf{x}} \in \bar{\Omega}$:

$$u^l(\mathbf{x}, \bar{\mathbf{x}}) \approx L_{\bar{\mathbf{x}}} u(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\bar{\mathbf{x}}),$$

where $u^l(\mathbf{x}, \bar{\mathbf{x}}) = \begin{cases} u(\mathbf{x}) & \forall \mathbf{x} \in \bar{\Omega}, |\mathbf{x} - \bar{\mathbf{x}}| < \rho \\ 0 & \text{otherwise} \end{cases}$ and the operator $L_{\bar{\mathbf{x}}}$ being a certain mapping. The vector $\mathbf{p}(\mathbf{x})$ is chosen according to Eq. 2.1 to be the complete basis in dimension d with consistency n . In order that the local approximation is the best approximation of u in a certain least-squares sense, the unknown coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ is selected to minimize the following weighted least-squares discrete L_2 error norm. That is, the coefficient vector $\mathbf{a}(\bar{\mathbf{x}})$ is selected to satisfy the condition $J_{\bar{\mathbf{x}}}(\mathbf{a}(\bar{\mathbf{x}})) \leq J_{\bar{\mathbf{x}}}(\mathbf{b})$ for all $\mathbf{b} \neq \mathbf{a} \in \mathbb{R}^k$ with

$$\begin{aligned} J_{\bar{\mathbf{x}}}(\mathbf{a}) &= \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) [L_{\bar{\mathbf{x}}} u(\mathbf{x}) - u^l(\mathbf{x}, \bar{\mathbf{x}})]^2 \\ &= \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) [\mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\bar{\mathbf{x}}) - u_i]^2. \end{aligned}$$

\mathbf{x}_i refers to the position of the N nodes within the domain which is discussed separately in subsection 4.4. The weighting function $w(\mathbf{x} - \mathbf{x}_i)$ plays an important role in the context of MMs which is worked out in subsection 4.5. It is defined on small supports $\tilde{\Omega}_i$ around each node thereby ensuring the locality of the approximation; the overlapping situation of the supports $\tilde{\Omega}_i$ within the domain is called *cover*. The weighting function may also be chosen individually for each node, then we write $w_i(\mathbf{x} - \mathbf{x}_i)$.

The functional $J_{\bar{\mathbf{x}}}(\mathbf{a})$ can be minimized by setting the derivative of $J_{\bar{\mathbf{x}}}(\mathbf{a})$ with respect to \mathbf{a} equal to zero, i.e. $\frac{\partial J_{\bar{\mathbf{x}}}(\mathbf{a})}{\partial \mathbf{a}} = 0$. The following system of k

equations results:

$$\begin{aligned} \frac{\partial J_{\bar{\mathbf{x}}}}{\partial a_1} = 0 : & \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) 2p_1(\mathbf{x}_i) [\mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\bar{\mathbf{x}}) - u_i] = 0 \\ \frac{\partial J_{\bar{\mathbf{x}}}}{\partial a_2} = 0 : & \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) 2p_2(\mathbf{x}_i) [\mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\bar{\mathbf{x}}) - u_i] = 0 \\ & \vdots \\ \frac{\partial J_{\bar{\mathbf{x}}}}{\partial a_k} = 0 : & \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) 2p_k(\mathbf{x}_i) [\mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\bar{\mathbf{x}}) - u_i] = 0. \end{aligned}$$

This is in vector notation

$$\begin{aligned} \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) 2\mathbf{p}(\mathbf{x}_i) [\mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\bar{\mathbf{x}}) - u_i] &= \mathbf{0} \\ 2 \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\bar{\mathbf{x}}) - w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) u_i &= \mathbf{0}. \end{aligned}$$

Eliminating the constant factor and separating the right hand side gives

$$\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\bar{\mathbf{x}}) = \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) u_i.$$

Solving this for $\mathbf{a}(\bar{\mathbf{x}})$ and then replacing $\mathbf{a}(\bar{\mathbf{x}})$ in the local approximation leads to

$$\begin{aligned} L_{\bar{\mathbf{x}}} u(\mathbf{x}) &= \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\bar{\mathbf{x}}) \\ &= \mathbf{p}^T(\mathbf{x}) \left[\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \right]^{-1} \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) u_i. \end{aligned}$$

In order to extend this local approximation to the whole domain, the so-called moving-procedure is introduced to achieve a global approximation. Since the point $\bar{\mathbf{x}}$ can be arbitrary chosen, one can let it “move” over the whole domain, $\bar{\mathbf{x}} \rightarrow \mathbf{x}$, which leads to the global approximation of $u(\mathbf{x})$ [99]. Mathematically a global approximation operator G is introduced with

$$u(\mathbf{x}) \approx Gu(\mathbf{x}) = u^h(\mathbf{x}),$$

where the operator G is another mapping, defined as $Gu(\mathbf{x}) = \lim_{\bar{\mathbf{x}} \rightarrow \mathbf{x}} L_{\bar{\mathbf{x}}} u(\mathbf{x})$ and can be interpreted as the globalization of the local approximation operator $L_{\bar{\mathbf{x}}}$ through the moving process [99]. Finally we obtain

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \left[\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \right]^{-1} \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) u_i. \quad (4.1)$$

This can be written shortly as:

$$u^h(\mathbf{x}) = Gu(\mathbf{x}) = \underbrace{\mathbf{p}^T(\mathbf{x})}_{(1 \times k)} \cdot \underbrace{[\mathbf{M}(\mathbf{x})]^{-1}}_{(k \times k)} \cdot \underbrace{\mathbf{B}(\mathbf{x})}_{(k \times N)} \cdot \underbrace{\mathbf{u}}_{(N \times 1)}$$

with

$$\mathbf{M}(\mathbf{x}) = \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i)$$

and

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} w(\mathbf{x} - \mathbf{x}_1) \mathbf{p}(\mathbf{x}_1) & w(\mathbf{x} - \mathbf{x}_2) \mathbf{p}(\mathbf{x}_2) & \dots & w(\mathbf{x} - \mathbf{x}_N) \mathbf{p}(\mathbf{x}_N) \end{bmatrix}.$$

The matrix $\mathbf{M}(\mathbf{x})$ in this expression is often called *moment matrix*; it is of size $k \times k$, i.e. of the same size than the complete basis vector $\mathbf{p}(\mathbf{x})$. This matrix has to be inverted wherever the MLS shape functions are to be evaluated. Obviously, for a higher desired order of consistency and thus higher values of k , this matrix inversion is not of negligible computing time due to the large number of evaluation points (=integration points in general) which are possibly involved.

Taking the viewpoint of an approximation of the form

$$u^h(\mathbf{x}) = \sum_{i=1}^N \Phi_i(\mathbf{x}) u_i = \Phi^T(\mathbf{x}) \mathbf{u},$$

we can immediately write for the shape functions

$$\Phi^T(\mathbf{x}) = \underbrace{\mathbf{p}^T(\mathbf{x}) [\mathbf{M}(\mathbf{x})]^{-1} \mathbf{B}(\mathbf{x})}_{(1 \times N)}$$

and thus for one certain shape function Φ_i at a point \mathbf{x}

$$\Phi_i(\mathbf{x}) = \underbrace{\mathbf{p}^T(\mathbf{x}) [\mathbf{M}(\mathbf{x})]^{-1} w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i)}_{(1 \times 1)}.$$

These shape functions fulfill the consistency requirements of order n and hence build a partition of unity of order n . It can easily be proved that functions of the basis $\mathbf{p}(\mathbf{x})$ are found exactly by the MLS approximation, see e.g. [17]. It is—at least for $n > 2$ —practically impossible to write down the shape functions in an explicit way, i.e. without the matrix inversion. Thus, we can

evaluate shape functions at arbitrary many points, but without knowing the shape functions explicitly. In the literature this is sometimes called “evaluating a function digitally”, as we do not know it in an explicit continuous (“analogous”) form.

It is important to note that any linear combination of the basis functions will indeed lead to the same shape functions, see proof e.g. in [57]. According to this, any translated and scaled basis can be used, leading to the same shape functions. This will be of importance for a better conditioning of the moment matrix, see subsection 4.6.

For a theoretical analysis of the MLS interpolants see [84].

The first derivatives of the MLS shape functions follow according to the product rule as

$$\begin{aligned} \Phi_{,k}^T(\mathbf{x}) &= \mathbf{p}_{,k}^T \mathbf{M}^{-1} \mathbf{B} + \\ &\quad \mathbf{p}^T \mathbf{M}_{,k}^{-1} \mathbf{B} + \\ &\quad \mathbf{p}^T \mathbf{M}^{-1} \mathbf{B}_{,k}, \end{aligned} \quad (4.2)$$

with $\mathbf{M}_{,k}^{-1} = -\mathbf{M}^{-1} \mathbf{M}_{,k} \mathbf{M}^{-1}$. The second derivatives are

$$\begin{aligned} \Phi_{,kl}^T(\mathbf{x}) &= \mathbf{p}_{,kl}^T \mathbf{M}^{-1} \mathbf{B} + \mathbf{p}_{,k}^T \mathbf{M}_{,l}^{-1} \mathbf{B} + \mathbf{p}_{,l}^T \mathbf{M}^{-1} \mathbf{B}_{,k} + \\ &\quad \mathbf{p}_{,l}^T \mathbf{M}_{,k}^{-1} \mathbf{B} + \mathbf{p}^T \mathbf{M}_{,kl}^{-1} \mathbf{B} + \mathbf{p}^T \mathbf{M}_{,k}^{-1} \mathbf{B}_{,l} + \\ &\quad \mathbf{p}_{,l}^T \mathbf{M}^{-1} \mathbf{B}_{,k} + \mathbf{p}^T \mathbf{M}_{,l}^{-1} \mathbf{B}_{,k} + \mathbf{p}^T \mathbf{M}^{-1} \mathbf{B}_{,kl}, \end{aligned} \quad (4.3)$$

with $\mathbf{M}_{,kl}^{-1} = \mathbf{M}^{-1} \mathbf{M}_{,l} \mathbf{M}^{-1} \mathbf{M}_{,k} \mathbf{M}^{-1} - \mathbf{M}^{-1} \mathbf{M}_{,kl} \mathbf{M}^{-1} + \mathbf{M}^{-1} \mathbf{M}_{,k} \mathbf{M}^{-1} \mathbf{M}_{,l} \mathbf{M}^{-1}$. In [16] Belytschko *et al.* propose an efficient way by means of a LU decomposition of the $k \times k$ system of equations to compute the derivatives of the MLS shape functions.

As an example, Fig. 2 shows shape functions and their derivatives in a one-dimensional domain $\Omega = [0, 1]$ with 11 equally distributed nodes. The weighting functions—discussed in detail in subsection 4.5—have a dilatation parameter of $\rho = 3 \cdot \Delta x = 0.3$. The following important properties can be seen:

- The dashed line in the upper picture shows that the sum of the shape functions $\sum_i \Phi_i(\mathbf{x})$ equals 1 in the whole domain, thus $\{\Phi_i\}$ builds a PU. The derivatives of the MLS-PU build Partition of Nullities (PNs), i.e. $\sum_i \frac{\partial \Phi_i(\mathbf{x})}{\partial x} = \sum_i \frac{\partial^2 \Phi_i(\mathbf{x})}{\partial x^2} = 0$.

- The rational shape functions themselves are smooth and can still be regarded to be rather polynomial-like, but the derivatives tend to have a more and more non-polynomial character. This will cause problems in integrating the integral expressions of the weak form, see subsection 6.2. Furthermore, the effort to evaluate the MLS shape function at each integration point might not be small as a matrix inversion is involved.
- The shape functions are not interpolating, i.e. they do not possess the Kronecker delta property. That means, at every node, there is more than one shape function $\neq 0$. Thus the computed values of a meshfree approximation are *not* nodal values. Due to this fact, they are sometimes called “fictitious values”. To have the real values of the sought function at a point, all influences of shape functions which are non-zero here have to be added up. The non-interpolant character makes imposition of essential boundary conditions difficult (see subsection 6.1). The lack of Kronecker Delta property is also a source of difficulties in error analysis of MMs for solving Dirichlet boundary value problems [59].

4.2.2 Deduction by Taylor Series Expansion

We can use a different starting point for the deduction of the MLS functions. A Taylor series expansion is the standard way to prove consistency of a certain order and we can —the other way around— use it to construct a consistent approximation. If we want to approximate a function $u(\mathbf{x})$ in the form

$$u^h(\mathbf{x}) = \sum_{i=1}^N \Phi_i(\mathbf{x}) u_i$$

we can evaluate a Taylor series expansion for any point $u_i = u(\mathbf{x}_i)$ as

$$u(\mathbf{x}_i) = \sum_{|\alpha|=0}^{\infty} \frac{(\mathbf{x}_i - \mathbf{x})^\alpha}{|\alpha|!} D^\alpha u(\mathbf{x}).$$

$\Phi_i(\mathbf{x})$ is chosen to be $\Phi_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i)$, which can be interpreted as a localized polynomial approximation. For computational reasons we write $(\mathbf{x}_i - \mathbf{x})$ as the argument of \mathbf{p} instead of (\mathbf{x}) . As \mathbf{p} only builds the basis of our approximation, there is no loss in generality due to this “shifting”. Inserting

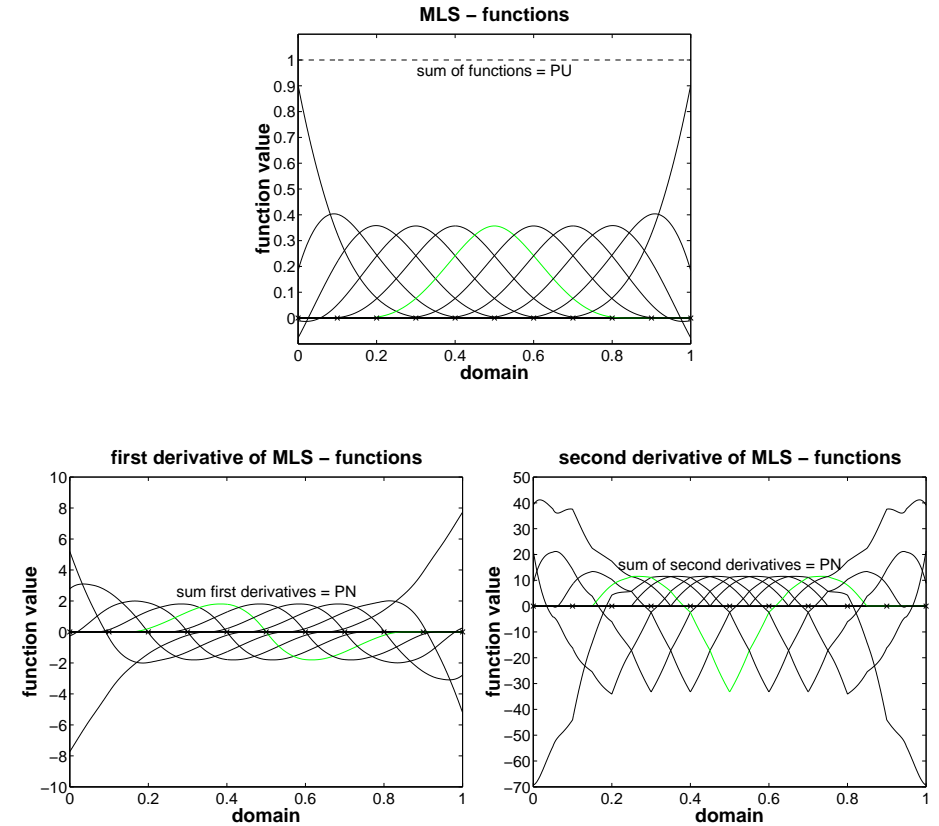


Figure 2: Partition of Unity functions and derivatives constructed with the MLS technique.

this into the approximation and then multiplying out leads to

$$\begin{aligned}
u^h(\mathbf{x}) &= \sum_{i=1}^N \Phi_i(\mathbf{x}) u_i \\
&= \sum_{i=1}^N \left[\left(\mathbf{p}^T(\mathbf{x}_i - \mathbf{x}) \mathbf{a}(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) \right) \left(\sum_{|\alpha|=0}^{\infty} \frac{(\mathbf{x}_i - \mathbf{x})^\alpha}{|\alpha|!} D^\alpha u(\mathbf{x}) \right) \right] \\
&= \sum_{i=1}^N \left[\left((\mathbf{x}_i - \mathbf{x})^{\alpha^1} a_1(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) + (\mathbf{x}_i - \mathbf{x})^{\alpha^2} a_2(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) + \right. \right. \\
&\quad \left. \left. \dots + (\mathbf{x}_i - \mathbf{x})^{\alpha^k} a_k(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) \right) \right. \\
&\quad \left(\frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^1}}{|\alpha^1|!} D^{\alpha^1} u(\mathbf{x}) + \frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^2}}{|\alpha^2|!} D^{\alpha^2} u(\mathbf{x}) + \right. \\
&\quad \left. \left. \dots + \frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^k}}{|\alpha^k|!} D^{\alpha^k} u(\mathbf{x}) + \dots \right) \right].
\end{aligned}$$

Comparing the coefficients on the left and on the right hand side shows that all terms on the right hand side referring to the derivatives of $u(\mathbf{x})$ must cancel out. If this could be fulfilled, the exact solution could be reached, however, in general an error term will remain. Our vector of unknowns consists of k components (k depends on the dimension and consistency) and so k equations can be derived out of the above expression. Note that $|\alpha^1| = 0$ and thus $\alpha^1 = (0, \dots, 0)$ and thus $D^{\alpha^1} u(\mathbf{x}) = u(\mathbf{x})$.

$$\begin{aligned}
1u^h(\mathbf{x}) &= \underbrace{1D^{\alpha^1} u(\mathbf{x})}_{\text{equation 1}} + \underbrace{0D^{\alpha^2} u(\mathbf{x})}_{\text{equation 2}} + \dots + \underbrace{0D^{\alpha^k} u(\mathbf{x})}_{\text{equation } k} + \text{error} \\
1u^h(\mathbf{x}) &= 1u(\mathbf{x}) + 0 + \dots + 0 + \text{error} \\
u^h(\mathbf{x}) &= u(\mathbf{x}) + \text{error}
\end{aligned}$$

A system of k equations follows:

$$\begin{aligned}
\sum_{i=1}^N \left((\mathbf{x}_i - \mathbf{x})^{\alpha^1} a_1 w(\mathbf{x} - \mathbf{x}_i) + \dots + (\mathbf{x}_i - \mathbf{x})^{\alpha^k} a_k w(\mathbf{x} - \mathbf{x}_i) \right) \frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^1}}{|\alpha^1|!} &= 1 \\
\sum_{i=1}^N \left((\mathbf{x}_i - \mathbf{x})^{\alpha^1} a_1 w(\mathbf{x} - \mathbf{x}_i) + \dots + (\mathbf{x}_i - \mathbf{x})^{\alpha^k} a_k w(\mathbf{x} - \mathbf{x}_i) \right) \frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^2}}{|\alpha^2|!} &= 0 \\
&\vdots \\
\sum_{i=1}^N \left((\mathbf{x}_i - \mathbf{x})^{\alpha^1} a_1 w(\mathbf{x} - \mathbf{x}_i) + \dots + (\mathbf{x}_i - \mathbf{x})^{\alpha^k} a_k w(\mathbf{x} - \mathbf{x}_i) \right) \frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^k}}{|\alpha^k|!} &= 0.
\end{aligned}$$

Writing $|\alpha^1|! = 0! = 1$ and neglecting all other $|\alpha^k|!$ terms as constants in homogenous equations and rearranging so that the vector of unknowns is extracted gives in matrix-vector notation

$$\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i - \mathbf{x}) \mathbf{p}^T(\mathbf{x}_i - \mathbf{x}) \mathbf{a}(\mathbf{x}) = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{p}(0).$$

Solving this for $\mathbf{a}(\mathbf{x})$ and inserting the result into the approximation finally gives

$$\begin{aligned}
u^h(\mathbf{x}) &= \sum_{i=1}^N \Phi_i(\mathbf{x}) u_i \\
&= \sum_{i=1}^N \mathbf{p}^T(\mathbf{x}_i - \mathbf{x}) \mathbf{a}(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) u_i \\
&= \sum_{i=1}^N \mathbf{p}^T(\mathbf{x}_i - \mathbf{x}) \left[\sum_{i=1}^N w \mathbf{p}(\mathbf{x}_i - \mathbf{x}) \mathbf{p}^T(\mathbf{x}_i - \mathbf{x}) \right]^{-1} \mathbf{p}(0) w u_i \\
&= \sum_{i=1}^N \mathbf{p}^T(0) \left[\sum_{i=1}^N w \mathbf{p}(\mathbf{x}_i - \mathbf{x}) \mathbf{p}^T(\mathbf{x}_i - \mathbf{x}) \right]^{-1} \mathbf{p}(\mathbf{x}_i - \mathbf{x}) w u_i.
\end{aligned}$$

We can now shift the basis another time by adding $+\mathbf{x}$ to all arguments which

leads to

$$\begin{aligned} u^h(\mathbf{x}) &= \sum_{i=1}^N \mathbf{p}^T(\mathbf{x}) \left[\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \right]^{-1} \mathbf{p}(\mathbf{x}_i) w(\mathbf{x} - \mathbf{x}_i) u_i \\ &= \mathbf{p}^T(\mathbf{x}) \left[\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \right]^{-1} \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) u_i, \end{aligned}$$

which is exactly the expression obtained via the MLS procedure of the previous subsection.

4.2.3 Deduction by Direct Imposition of the Consistency Conditions

It is shown in subsection 2.5 that functions of n th order consistency satisfy the following equations:

$$\sum_i \Phi_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i) = \mathbf{p}(\mathbf{x}), \quad (4.4)$$

which is a system of k equations. In this subsection the functions Φ_i are determined by directly fulfilling these k equations. Φ_i has the form $\Phi_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i)$ where we write for computational reasons instead of $\mathbf{p}(\mathbf{x})$ the shifted basis

$$\mathbf{p}^*(\mathbf{x}) = \begin{pmatrix} p_1(\mathbf{x}) \\ p_2(\mathbf{x}_i) - p_2(\mathbf{x}) \\ p_3(\mathbf{x}_i) - p_3(\mathbf{x}) \\ \vdots \\ p_k(\mathbf{x}_i) - p_k(\mathbf{x}) \end{pmatrix}.$$

The first line in our system of k equations is $\sum_i \Phi_i(\mathbf{x}) p_1(\mathbf{x}_i) = p_1(\mathbf{x})$ and due to $p_1(\mathbf{x}) = 1$ (see subsection 2.4) and thus $p_1(\mathbf{x}_i) = 1$ as well, we can write $\sum_i \Phi_i(\mathbf{x}) 1 = 1$. Multiplying this with $\mathbf{p}(\mathbf{x})$ on both side gives

$$\sum_i \Phi_i(\mathbf{x}) \mathbf{p}(\mathbf{x}) = \mathbf{p}(\mathbf{x}). \quad (4.5)$$

Lines 2, 3, \dots , k of Eq. 4.5 are subtracted from the corresponding lines in Eq. 4.4 which results into

$$\begin{aligned} \sum_i \Phi_i(\mathbf{x}) \begin{pmatrix} p_1(\mathbf{x}) \\ p_2(\mathbf{x}_i) - p_2(\mathbf{x}) \\ p_3(\mathbf{x}_i) - p_3(\mathbf{x}) \\ \vdots \\ p_k(\mathbf{x}_i) - p_k(\mathbf{x}) \end{pmatrix} &= \begin{pmatrix} p_1(\mathbf{x}) \\ p_2(\mathbf{x}) - p_2(\mathbf{x}) \\ p_3(\mathbf{x}) - p_3(\mathbf{x}) \\ \vdots \\ p_k(\mathbf{x}) - p_k(\mathbf{x}) \end{pmatrix} \\ \sum_i \Phi_i(\mathbf{x}) \mathbf{p}^*(\mathbf{x}) &= \mathbf{p}^{**}(\mathbf{x}). \end{aligned}$$

Clearly, $\mathbf{p}^{**}(\mathbf{x})$ reduces to $(0, 0, \dots, 1)^T$. Solving the resulting system of equations after inserting $\Phi_i(\mathbf{x}) = \mathbf{p}^{*T}(\mathbf{x}) \mathbf{a}(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i)$ for the unknowns $\mathbf{a}(\mathbf{x})$ gives

$$\begin{aligned} \sum_i \mathbf{p}^{*T}(\mathbf{x}) \mathbf{a}(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}^*(\mathbf{x}) &= \mathbf{p}^{**}(\mathbf{x}) \\ \sum_i w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}^*(\mathbf{x}) \mathbf{p}^{*T}(\mathbf{x}) \mathbf{a}(\mathbf{x}) &= \mathbf{p}^{**}(\mathbf{x}) \\ \Rightarrow \mathbf{a}(\mathbf{x}) &= \left[\sum_i w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}^*(\mathbf{x}) \mathbf{p}^{*T}(\mathbf{x}) \right]^{-1} \mathbf{p}^{**}(\mathbf{x}). \end{aligned}$$

Thus, for the approximation we obtain

$$\begin{aligned} u^h(\mathbf{x}) &= \sum_{i=1}^N \Phi_i(\mathbf{x}) u_i \\ &= \sum_{i=1}^N \mathbf{p}^{*T}(\mathbf{x}) \mathbf{a}(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) u_i \\ &= \sum_{i=1}^N \mathbf{p}^{*T}(\mathbf{x}) \left[\sum_i w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}^*(\mathbf{x}) \mathbf{p}^{*T}(\mathbf{x}) \right]^{-1} \mathbf{p}^{**}(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) u_i. \end{aligned}$$

One may shift the basis with $(0, +p_2(\mathbf{x}), \dots, +p_k(\mathbf{x}))^T$ which gives $\mathbf{p}^*(\mathbf{x}) \rightarrow \mathbf{p}(\mathbf{x}_i)$ and $\mathbf{p}^{**}(\mathbf{x}) \rightarrow \mathbf{p}(\mathbf{x})$ and thus one obtains after some rearranging

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \left[\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \right]^{-1} \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) u_i.$$

This is exactly the same approximation as found with the MLS approach shown in subsection 4.2.1.

4.2.4 Generalized Moving Least-Squares (GMLS)

With the GMLS it is possible to treat the derivatives of a function as independent functions. This can for example be important for the solution of 4th order boundary value problems (e.g. analysis of thin beams), where displacement and slope BCs might be imposed at the same point (which is not possible in 2nd order problems) [2]. In this case, not only the values u_i of a function are unknowns but also their derivatives up to a certain degree. The local approximation is then carried out using the following weighted discrete H^l error norm instead of the above used L_2 error norm:

$$J_{\bar{\mathbf{x}}}^{(l)}(\mathbf{a}) = \sum_{j=1}^q \sum_{i=1}^N w^{(\alpha^j)}(\mathbf{x} - \mathbf{x}_i) \left[D^{\alpha^j} \mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\bar{\mathbf{x}}) - D^{\alpha^j} u(\mathbf{x}_i) \right]^2.$$

The unknown vector $\mathbf{a}(\bar{\mathbf{x}})$ is again obtained by minimizing this norm as in the standard MLS by setting $\frac{\partial J_{\bar{\mathbf{x}}}(\mathbf{a})}{\partial \mathbf{a}} = 0$, that is

$$2 \sum_{j=1}^q \sum_{i=1}^N w^{(\alpha^j)}(\mathbf{x} - \mathbf{x}_i) D^{\alpha^j} \mathbf{p}(\mathbf{x}_i) D^{\alpha^j} \mathbf{p}^T(\mathbf{x}_i) \mathbf{a}(\bar{\mathbf{x}}) - w^{(\alpha^j)}(\mathbf{x} - \mathbf{x}_i) D^{\alpha^j} \mathbf{p}(\mathbf{x}_i) D^{\alpha^j} u_i = \mathbf{0}.$$

The MLS system of equations is still of the same order $k \times k$ and the extra effort lies in building q times the sum over all points, with $q \leq k$ being the number of derivatives which shall be included in the approximation as unknowns [2].

Without repeating the details of the moving procedure, the approximation will —after solving the system for the unknown $\mathbf{a}(\mathbf{x})$ — be of the form

$$u^h(\mathbf{x}) = \sum_{j=1}^q \left(\mathbf{p}^T(\mathbf{x}) \left[\sum_{i=1}^N w^{(\alpha^j)}(\mathbf{x} - \mathbf{x}_i) D^{\alpha^j} \mathbf{p}(\mathbf{x}_i) D^{\alpha^j} \mathbf{p}^T(\mathbf{x}_i) \right]^{-1} \sum_{i=1}^N w^{(\alpha^j)}(\mathbf{x} - \mathbf{x}_i) D^{\alpha^j} \mathbf{p}(\mathbf{x}_i) D^{\alpha^j} u_i \right).$$

4.2.5 Relation to Shepard's Method

In 1968 Shepard introduced in the context of data interpolation the following approximation:

$$u^h(\mathbf{x}) = \sum_{i=1}^N \Phi_i(\mathbf{x}) u_i,$$

with

$$\Phi_i(\mathbf{x}) = \frac{w(\mathbf{x} - \mathbf{x}_i)}{\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i)}.$$

This can be interpreted as an inverse distance weighting. It can readily be shown that this forms a PU: $\sum_{i=1}^N \Phi_i(\mathbf{x}) = \frac{\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i)}{\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i)} = 1$.

It can also be shown that the MLS with the polynomial basis of 0-th order consistency, hence $\mathbf{p}(\mathbf{x}) = [1]$, leads to the same result:

$$\begin{aligned} u^h(\mathbf{x}) &= \mathbf{p}^T(\mathbf{x}) \left[\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \right]^{-1} \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) u_i \\ &= 1 \left[\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) 1 \cdot 1 \right]^{-1} \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) 1 u_i \\ &= \sum_{i=1}^N \frac{w(\mathbf{x} - \mathbf{x}_i)}{\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i)} u_i. \end{aligned}$$

Thus, Shepard's method is clearly a subcase of the MLS procedure with consistency of 0-th order. Using the Shepard method to construct a meshfree PU has the important advantage of low computational cost and simplicity of computation. The problem is clearly the low order of consistency which make the Shepard PU fail for the solution of even second order boundary value problems. But, it shall be mentioned that ways have been shown to construct a linear-precision (=first order consistent) PU based on the Shepard's method with only small computational extra effort [79]. Furthermore, it is mentioned in [17] that in fact Shepard functions have been used for the simulation of second-order PDEs, showing that consistency is sufficient (stability provided) but may not be necessary for convergence.

Another approach is to introduce unknowns of the derivatives and then use “star nodes” to determine the derivative data. The closest nodes are chosen as star nodes and there must be at least two star nodes in order to be able to construct a basis with linear precision. In this case, the problem is that there are more unknowns (having different physical meanings) and the undesirable effect that it may well lead to numerical difficulties due to the conditioning of the global matrices [79].

There is also a way to compute a first order consistent PU based on Shepard's method only using one type of nodal parameter, thus only the values of the sought function at nodes shall occur as degrees of freedom and the number of

degrees of freedom is not increased by including also derivatives of the solution as unknowns. This has been presented by Krysl and Belytschko in [79]. Here, the nodal functions are sought as linear combinations of the nodal parameters at the central node and a set of near-by nodes, called the star-nodes. This method is very efficient, in fact the cost of the precision enhancement constitutes only a small fraction of the cost involved in the construction of the Shepard basis.

4.2.6 Relation to other Least-Squares Schemes

Oñate *et al.* pointed out in [115] that any least-squares scheme can be used for an approximation, hence for obtaining certain shape functions. The basic idea is always to minimize the sum of the square distances of the error at any point weighted with a certain function w , hence to minimize

$$\begin{aligned} J &= \sum_{i=1}^N w (u^h(\mathbf{x}_i) - u(\mathbf{x}_i))^2 \\ &= \sum_{i=1}^N w (\mathbf{p}^T(\mathbf{x}_i) \mathbf{a} - u_i)^2. \end{aligned}$$

All least-squares schemes can be motivated from this starting point [115], as can be seen in the following, see also Fig. 3.

$w = 1$: The Standard Least-Squares method (LSQ) results, where the functional that has to be minimized becomes $J = \sum_{i=1}^N (\mathbf{p}^T(\mathbf{x}_i) \mathbf{a} - u_i)^2$. The main drawback of the LSQ approach is that the approximation rapidly deteriorates if the number of points N used, largely exceeds that of the k polynomial terms in the basis \mathbf{p} . From the minimization the system of equations for \mathbf{a} becomes

$$\sum_{i=1}^N \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \mathbf{a} = \sum_{i=1}^N \mathbf{p}(\mathbf{x}_i) u_i.$$

The unknowns \mathbf{a} take one certain value which can be inserted in the approximation.

$w = w_j(\mathbf{x}_i)$: Choosing w like this leads to the Fixed Least-Squares method (FLS). For the approximation of u at a certain point \mathbf{x} a fixed weighting function w_j is chosen due to some criterion. This function w_j

has its highest value at \mathbf{x}_j which is near \mathbf{x} . It is zero outside a region Ω_j around \mathbf{x}_j . w_j might not be the only function which is non-zero at \mathbf{x} , and the choice of the “belonging” weighting function is not always easy. But more than one weighting function would clearly lead to a multivalued interpolation. The system of equations is

$$\sum_{i=1}^N w_j(\mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \mathbf{a} = \sum_{i=1}^N w_j(\mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) u_i.$$

The unknowns \mathbf{a} are constant values as long as one certain w_j is chosen at a point \mathbf{x} , otherwise different \mathbf{a} would result leading to an inadmissible multivalued approximation. For *different* positions \mathbf{x} one has to choose other w_j (of course, only *one* w_j for each position \mathbf{x}) which can introduce jumps to the approximation.

$w = w(\mathbf{x} - \mathbf{x}_i)$: In the Moving Least-Squares (MLS) approach the weighting function w is defined in shape and size and is translated over the domain so that it takes the maximum value at an arbitrary point \mathbf{x} , where the unknown of the function u is to be evaluated. Now, for every point the following functional $J_i = \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) (\mathbf{p}^T(\mathbf{x}_i) \mathbf{a} - u_i)^2$ is minimized. w can in general change its shape and span depending on the position of point \mathbf{x} . For the system results

$$\sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \mathbf{a} = \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) u_i.$$

The parameters \mathbf{a} are no longer constants but vary continuously with position \mathbf{x} so one should rather write $\mathbf{a}(\mathbf{x})$. The inversion of matrices is required at every point where u is to be evaluated.

$w = w_i(\mathbf{x})$: This leads to the Multiple Fixed Least-Squares (MFLS) method where the functional $J_i = \sum_{i=1}^N w_i(\mathbf{x}) (\mathbf{p}^T(\mathbf{x}_i) \mathbf{a} - u_i)^2$ is minimized. The name stems from the fact that multiple fixed weight functions w_i each having their maximum at position \mathbf{x}_i are considered at point \mathbf{x} . The definitions of shape functions due to this approach is still unique although more than one w_j has an influence (in contrast to what was claimed for the FLS). But due to the specific “averaging” of all weighting functions having influence at point

\mathbf{x} unique shape functions result. The system is

$$\sum_{i=1}^N w_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \mathbf{a} = \sum_{i=1}^N w_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i) u_i.$$

A graphical representation of the relations between the different least-squares schemes is depicted in Fig. 3. A typical choice for w is the Gaussian function or any Gaussian-like function, this is discussed in further detail in subsection 4.5. Of course, $N \geq k$ is always required in the sampling region and if equality occurs no effect of the weighting is present and the interpolation is the same as in the LSQ scheme [115]. Any of the shown least-squares schemes can be used for the construction of meshfree shape functions. However, in practical use the MLS shape functions are most often chosen and have therefore been presented in very detail.

If the weight function w depends on a mesh, it is possible to obtain the classical mesh-based shape functions via a least-squares procedure. This is further pointed out in subsection 5.2.

4.3 Reproducing Kernel Particle Method (RKPM)

The RKPM is motivated by the theory of wavelets where functions are represented by a combination of the dilatation and translation of a single wavelet. Reproducing kernel methods are in general a class of operators that reproduce the function itself through integration over the domain [97]. Here, we are interested in an integral transform of the type

$$u^h(\mathbf{x}) = \int_{\Omega_y} K(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) d\Omega_y.$$

Clearly, if the kernel $K(\mathbf{x}, \mathbf{y})$ equals the Dirac function $\delta(\mathbf{x} - \mathbf{y})$, the function $u(\mathbf{x})$ will be reproduced exactly. It is important to note that the reproducing kernel method (RKM) is a *continuous* form of an approximation. However, for the evaluation of such an integral in practice, the RKM has to be discretized, hence

$$u^h(\mathbf{x}) = \sum_{i=1}^N K(\mathbf{x}_i - \mathbf{x}, \mathbf{x}) u_i \Delta V_i.$$

This *discrete* version is then called reproducing kernel particle method (RKPM).

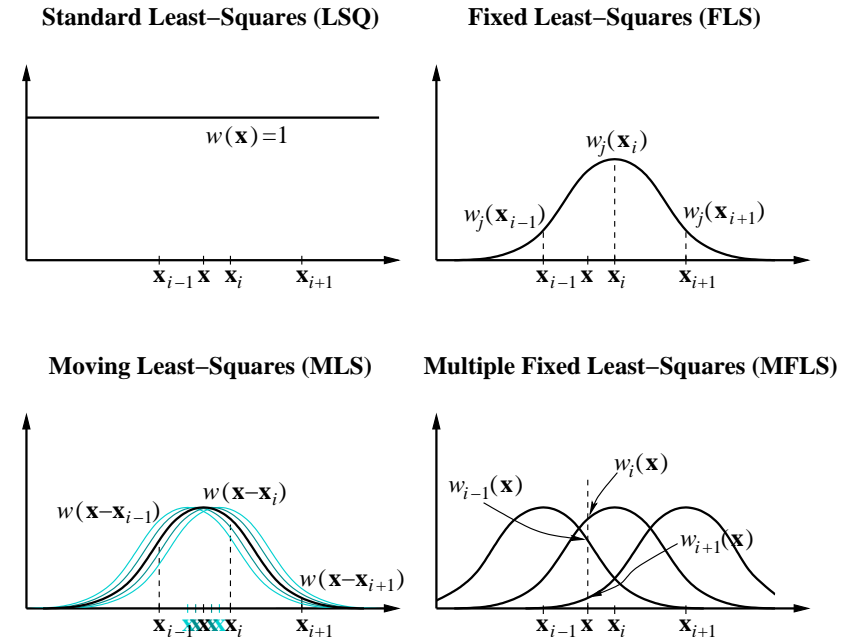


Figure 3: Different least-squares schemes.

The kernel $K(\mathbf{x}, \mathbf{y})$ shall be constructed such that it reproduces polynomials of order n exactly, i.e. it leads to an n -th order consistent approximation. We write

$$\begin{aligned} u^h(\mathbf{x}) &= \int_{\Omega_y} K(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) d\Omega_y \\ &= \int_{\Omega_y} C(\mathbf{x}, \mathbf{x} - \mathbf{y}) w(\mathbf{x} - \mathbf{y}) u(\mathbf{y}) d\Omega_y, \end{aligned}$$

where $w(\mathbf{x} - \mathbf{y})$ is a weighting function (in the context of the RKPM also called window function). If we had $K(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} - \mathbf{y})$ the approximation would not be able to fulfill the required consistency requirements (which is the drawback of wavelet and SPH method [94], see subsection 5.1). Therefore, the kernel $K(\mathbf{x} - \mathbf{y}) = w(\mathbf{x} - \mathbf{y})$ has been modified with a correction function $C(\mathbf{x}, \mathbf{y})$ so that it reproduces polynomials exactly, leading to $K(\mathbf{x}, \mathbf{y}) = C(\mathbf{x}, \mathbf{y}) w(\mathbf{x} - \mathbf{y})$. To define the modified kernel the correction function has to be determined such that the approximation is n -th order consistent. Several approaches are shown in the following.

1.) This approach has been proposed in [98]. Here, we want to represent a function $u(\mathbf{x})$ with the basis $\mathbf{p}(\mathbf{x})$, hence $u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}$ (remark: Li *et al.* write \mathbf{a} although in the result it becomes clear that it is not constant for changing \mathbf{x}). In order to determine the unknown coefficients \mathbf{a} both sides are multiplied by $\mathbf{p}(\mathbf{x})$ and an integral window transform is performed with respect to a window function $w(\mathbf{x} - \mathbf{y})$ to obtain

$$\begin{aligned} u(\mathbf{x}) &= \mathbf{p}^T(\mathbf{x}) \mathbf{a} \\ \mathbf{p}(\mathbf{x}) u(\mathbf{x}) &= \mathbf{p}(\mathbf{x}) \mathbf{p}^T(\mathbf{x}) \mathbf{a} \\ \int_{\Omega_y} \mathbf{p}(\mathbf{y}) w(\mathbf{x} - \mathbf{y}) u(\mathbf{y}) d\Omega_y &= \int_{\Omega_y} \mathbf{p}(\mathbf{y}) \mathbf{p}^T(\mathbf{y}) w(\mathbf{x} - \mathbf{y}) d\Omega_y \mathbf{a}. \end{aligned}$$

This is a system of equations for determining \mathbf{a} . Solving for \mathbf{a} and inserting this into $u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}$ gives finally

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \left[\int_{\Omega_y} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y}) \mathbf{p}^T(\mathbf{y}) d\Omega_y \right]^{-1} \int_{\Omega_y} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y}) u(\mathbf{y}) d\Omega_y.$$

2.) This approach uses the moving least-squares idea in a continuous way and was proposed in [33, 99]. We start with a local approximation $u(\mathbf{x}) \approx L_{\bar{\mathbf{x}}} u(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\bar{\mathbf{x}})$ (note, in the original papers this is chosen as $u^h(\mathbf{x}) =$

$\mathbf{p}^T \left(\frac{\mathbf{x} - \bar{\mathbf{x}}}{\rho} \right) \mathbf{a}(\bar{\mathbf{x}})$ for computational reasons, see subsection 4.6). A continuous localized error functional is introduced:

$$J(\mathbf{a}(\bar{\mathbf{x}})) = \int_{\Omega} w(\bar{\mathbf{x}} - \mathbf{x}) [u(\mathbf{x}) - \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\bar{\mathbf{x}})]^2 d\Omega,$$

which has to be minimized in order to determine the unknowns $\mathbf{a}(\bar{\mathbf{x}})$. In the same way as shown for the MLS in subsection 4.2.1 we take the derivative of the functional with respect to $\mathbf{a}(\bar{\mathbf{x}})$ and set this to zero in order to determine the minimum, that is

$$\frac{\partial J(\mathbf{a}(\bar{\mathbf{x}}))}{\partial \mathbf{a}(\bar{\mathbf{x}})} = \int_{\Omega_x} w(\bar{\mathbf{x}} - \mathbf{x}) 2\mathbf{p}(\mathbf{x}) [u(\mathbf{x}) - \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\bar{\mathbf{x}})] d\Omega_x = 0,$$

and after some rearranging the system becomes

$$\int_{\Omega_x} w(\bar{\mathbf{x}} - \mathbf{x}) \mathbf{p}(\mathbf{x}) \mathbf{p}^T(\mathbf{x}) d\Omega_x \mathbf{a}(\bar{\mathbf{x}}) = \int_{\Omega_x} w(\bar{\mathbf{x}} - \mathbf{x}) \mathbf{p}(\mathbf{x}) u(\mathbf{x}) d\Omega_x.$$

Solving for $\mathbf{a}(\bar{\mathbf{x}})$, inserting this into $L_{\bar{\mathbf{x}}} u(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\bar{\mathbf{x}})$ gives

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \left[\int_{\Omega_x} w(\bar{\mathbf{x}} - \mathbf{x}) \mathbf{p}(\mathbf{x}) \mathbf{p}^T(\mathbf{x}) d\Omega_x \right]^{-1} \int_{\Omega_x} w(\bar{\mathbf{x}} - \mathbf{x}) \mathbf{p}(\mathbf{x}) u(\mathbf{x}) d\Omega_x.$$

Note that \mathbf{x} is a dummy variable since integration is performed over Ω_x , and $\int_{\Omega_a} f(\mathbf{a}) d\Omega_a = \int_{\Omega_b} f(\mathbf{b}) d\Omega_b$. Thus \mathbf{x} may be replaced in the integrals by \mathbf{y} :

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \left[\int_{\Omega_y} w(\bar{\mathbf{x}} - \mathbf{y}) \mathbf{p}(\mathbf{y}) \mathbf{p}^T(\mathbf{y}) d\Omega_y \right]^{-1} \int_{\Omega_y} w(\bar{\mathbf{x}} - \mathbf{y}) \mathbf{p}(\mathbf{y}) u(\mathbf{y}) d\Omega_y.$$

Applying the 'moving procedure' $\bar{\mathbf{x}} \rightarrow \mathbf{x}$ as explained in subsection 4.2.1 finally gives

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \left[\int_{\Omega_y} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y}) \mathbf{p}^T(\mathbf{y}) d\Omega_y \right]^{-1} \int_{\Omega_y} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y}) u(\mathbf{y}) d\Omega_y.$$

3.) This approach works with help of a Taylor series expansion as done in [25]. It starts with

$$u^h(\mathbf{x}) = \int_{\Omega_y} C(\mathbf{x}, \mathbf{y}) w(\mathbf{x} - \mathbf{y}) u(\mathbf{y}) d\Omega_y.$$

For the correction function $C(\mathbf{x}, \mathbf{y})$ we assume $C(\mathbf{x}, \mathbf{y}) = \mathbf{p}^T(\mathbf{y} - \mathbf{x}) \mathbf{a}(\mathbf{x})$, where the basis functions \mathbf{p} are shifted for simplicity reasons, and for $u(\mathbf{y})$ we make the Taylor series expansion

$$u^h(\mathbf{x}) = \int_{\Omega_{\mathbf{y}}} \left[\mathbf{p}^T(\mathbf{y} - \mathbf{x}) \mathbf{a}(\mathbf{x}) w(\mathbf{x} - \mathbf{y}) \sum_{|\alpha|=0}^{\infty} \frac{(\mathbf{y} - \mathbf{x})^\alpha}{|\alpha|!} D^\alpha u(\mathbf{x}) \right] d\Omega_{\mathbf{y}}.$$

The following steps are identical as shown for the Taylor series expansion for the MLS in subsection 4.2.2. We insert $\mathbf{p}(\mathbf{y} - \mathbf{x}) = (\mathbf{y} - \mathbf{x})^\alpha$, multiply the expressions out and compare the coefficients of the terms $D^\alpha u(\mathbf{x})$. This leads to the following system of equations:

$$\begin{aligned} \int_{\Omega_{\mathbf{y}}} \left((\mathbf{y} - \mathbf{x})^{\alpha^1} a_1 w(\mathbf{x} - \mathbf{y}) + \dots + (\mathbf{y} - \mathbf{x})^{\alpha^k} a_k w(\mathbf{x} - \mathbf{y}) \right) \frac{(\mathbf{y} - \mathbf{x})^{\alpha^1}}{|\alpha^1|!} d\Omega_{\mathbf{y}} &= 1 \\ \int_{\Omega_{\mathbf{y}}} \left((\mathbf{y} - \mathbf{x})^{\alpha^1} a_1 w(\mathbf{x} - \mathbf{y}) + \dots + (\mathbf{y} - \mathbf{x})^{\alpha^k} a_k w(\mathbf{x} - \mathbf{y}) \right) \frac{(\mathbf{y} - \mathbf{x})^{\alpha^2}}{|\alpha^2|!} d\Omega_{\mathbf{y}} &= 0 \\ &\vdots \\ \int_{\Omega_{\mathbf{y}}} \left((\mathbf{y} - \mathbf{x})^{\alpha^1} a_1 w(\mathbf{x} - \mathbf{y}) + \dots + (\mathbf{y} - \mathbf{x})^{\alpha^k} a_k w(\mathbf{x} - \mathbf{y}) \right) \frac{(\mathbf{y} - \mathbf{x})^{\alpha^k}}{|\alpha^k|!} d\Omega_{\mathbf{y}} &= 0, \end{aligned}$$

and in vector notation

$$\int_{\Omega_{\mathbf{y}}} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y} - \mathbf{x}) \mathbf{p}^T(\mathbf{y} - \mathbf{x}) \mathbf{a}(\mathbf{x}) d\Omega_{\mathbf{y}} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{p}(0).$$

Solving for $\underline{a}(\mathbf{x})$ and inserting this into the correction function gives

$$C(\mathbf{x}, \mathbf{y}) = \mathbf{p}^T(\mathbf{y} - \mathbf{x}) \left[\int_{\Omega_{\mathbf{y}}} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y} - \mathbf{x}) \mathbf{p}^T(\mathbf{y} - \mathbf{x}) d\Omega_{\mathbf{y}} \right]^{-1} \mathbf{p}(0).$$

For the approximation follows

$$\begin{aligned} u^h(\mathbf{x}) &= \int_{\Omega_{\mathbf{y}}} C(\mathbf{x}, \mathbf{y}) w(\mathbf{x} - \mathbf{y}) u(\mathbf{y}) d\Omega_{\mathbf{y}} \\ &= \int_{\Omega_{\mathbf{y}}} \mathbf{p}^T(\mathbf{y} - \mathbf{x}) \left[\int_{\Omega_{\mathbf{y}}} w(\mathbf{y} - \mathbf{x}) \mathbf{p}(\mathbf{y} - \mathbf{x}) \mathbf{p}^T(\mathbf{y} - \mathbf{x}) d\Omega_{\mathbf{y}} \right]^{-1} \mathbf{p}(0) w u(\mathbf{y}) d\Omega_{\mathbf{y}} \\ &= \mathbf{p}^T(0) \left[\int_{\Omega_{\mathbf{y}}} w(\mathbf{y} - \mathbf{x}) \mathbf{p}(\mathbf{y} - \mathbf{x}) \mathbf{p}^T(\mathbf{y} - \mathbf{x}) d\Omega_{\mathbf{y}} \right]^{-1} \int_{\Omega_{\mathbf{y}}} \mathbf{p}(\mathbf{y} - \mathbf{x}) w u(\mathbf{y}) d\Omega_{\mathbf{y}}. \end{aligned}$$

After shifting the basis another time the final approximation is obtained as

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \left[\int_{\Omega_{\mathbf{y}}} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y}) \mathbf{p}^T(\mathbf{y}) d\Omega_{\mathbf{y}} \right]^{-1} \int_{\Omega_{\mathbf{y}}} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y}) u(\mathbf{y}) d\Omega_{\mathbf{y}}. \quad (4.6)$$

One can thus see that all three approaches of the RKM give the same resulting continuous approximations for $u(\mathbf{x})$. And also the similarities of the RKM and the MLS can be seen. The important difference is that the MLS uses discrete expressions (sums over a number of points), see Eq. 4.1, whereas in the RKM we have continuous integrals, see Eq. 4.6. For example the discrete moment matrix $\mathbf{M}(\mathbf{x})$ of the MLS is $\mathbf{M}(\mathbf{x}) = \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i)$ whereas the continuous moment matrix $\mathbf{M}(\mathbf{x})$ of the RKM is $\mathbf{M}(\mathbf{x}) = \int_{\Omega_{\mathbf{y}}} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y}) \mathbf{p}^T(\mathbf{y}) d\Omega_{\mathbf{y}}$.

The modified kernel $K(\mathbf{x}, \mathbf{y})$ fulfills consistency requirements up to order n . The correction function of the modified kernel can be identified as

$$\begin{aligned} C(\mathbf{x}, \mathbf{y}) &= \mathbf{p}^T(\mathbf{x}) \left[\int_{\Omega_{\mathbf{y}}} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y} - \mathbf{x}) \mathbf{p}^T(\mathbf{y} - \mathbf{x}) d\Omega_{\mathbf{y}} \right]^{-1} \mathbf{p}(\mathbf{y}) \\ &= \mathbf{p}^T(\mathbf{x}) [\mathbf{M}(\mathbf{x})]^{-1} \mathbf{p}(\mathbf{y}). \end{aligned}$$

This correction function most importantly takes boundary effects into account. Therefore, the correction function is sometimes referred to as boundary correction term [94]. Far away from the boundary the correction function plays almost no role [97, 98].

To evaluate the above continuous integral expressions numerical integration, thus discretization, is necessary. This step leads from the RKM to the RKPM. This has not yet directly the aim to evaluate the integrals of the weak form of the PDE, but more to yield shape functions to work with. To do this, an

admissible particle distribution (see subsection 4.6) has to be set up [99]. Then the integral can be approximated as

$$\begin{aligned} u^h(\mathbf{x}) &= \int_{\Omega_{\mathbf{y}}} C(\mathbf{x}, \mathbf{y}) w(\mathbf{x} - \mathbf{y}) u(\mathbf{y}) d\Omega_{\mathbf{y}} \\ &= \sum_{i=1}^N C(\mathbf{x}, \mathbf{x}_i) w(\mathbf{x} - \mathbf{x}_i) u_i \Delta V_i \\ &= \mathbf{p}^T(\mathbf{x}) [\mathbf{M}(\mathbf{x})]^{-1} \sum_{i=1}^N \mathbf{p}(\mathbf{x}_i) w(\mathbf{x} - \mathbf{x}_i) u_i \Delta V_i. \end{aligned} \quad (4.7)$$

Numerical integration is also required to evaluate the moment matrix $\mathbf{M}(\mathbf{x})$:

$$\begin{aligned} \mathbf{M}(\mathbf{x}) &= \int_{\Omega_{\mathbf{y}}} w(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y}) \mathbf{p}^T(\mathbf{y}) d\Omega_{\mathbf{y}} \\ &= \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \Delta V_i \end{aligned}$$

The choice of the integration weights ΔV_i , hence the influence of each particle in the evaluation of the integral or more descriptive the particle's lumped volume, is *not* prescribed. However, once a certain quadrature rule is chosen, it should be carried out through all the integrals consistently. The choice $\Delta V_i = 1$ leads to *exactly* the same RKPM approximation as the MLS approximation [99]; compare Eq. 4.1 and 4.7. The equivalence between MLS and RKPM is a remarkable result which unifies two methodologies with very different origins, it has also been discussed in [17] and [87]. Belytschko *et al.* claim in [17]:

Any kernel method in which the parent kernel is identical to the weight function of a MLS approximation and is rendered consistent by the same basis is identical. In other words, a discrete kernel approximation which is consistent must be identical to the related MLS approximation.

In [87], Li and Liu make the point that the use of a shifted basis $\mathbf{p}(\mathbf{x} - \mathbf{x}_i)$ instead of $\mathbf{p}(\mathbf{x})$ may not be fully equivalent in cases where other basis functions than monomials are used.

It should be mentioned that $\Delta V_i = 1$ cannot be called a suitable approximation of an integral *in general*. Consider the following example, where the

integral $\int_{\Omega} 1 d\Omega$ shall be performed with $\sum_{i=1}^N \Delta V_i$. Choosing $\Delta V_i = 1$ does not make sense in this case as the result $\sum_{i=1}^N \Delta V_i = \sum_{i=1}^N 1 = N$ will only be dependent of the number of integration points N without any relation to the integral itself.

In case of integrating the above RKPM expressions one can find that ΔV_i appears in two integrals. The one for the moment matrix $\mathbf{M}(\mathbf{x})$ is later inverted. So for constant $\Delta V_i = c$, $c \in \Re$ the same functions will result for any c . Hence, for constant ΔV_i it is not important whether or not the result is a measure of the domain ($\sum_{i=1}^N \Delta V_i = meas\{\Omega\}$). However, more suitable integration weights ΔV_i might be employed, with ΔV_i not being constant but dependent of the particle density, dilatation parameter etc. Then, other PU functions than the standard MLS shape functions are obtained.

To the author's knowledge no systematical studies with $\Delta V_i \neq 1$ have yet been published. However, we mention [1] as a paper, where experiences with different choices for ΔV_i have been made. There, "correct" volumes ΔV_i , $\Delta V_i = 1$ and $\Delta V_i = \text{random values}$ have been tested. It is mentioned that consistency of the resulting RKPM shape functions may be obtained in any of these three cases, but other values than $\Delta V_i = 1$ do not show advantages; for the random values the approximation properties clearly degrade. Therefore in the following, we only consider $\Delta V_i = 1$ where RKPM equals MLS, but keep in mind that there is a difference between RKPM and MLS (see [86, 94] for further information) which in practice seems to be of less importance.

Important aspects of the RKM and RKPM shall be pointed out in more detail in the following, where some citations of important statements are given.

Due to the discretization procedure error terms are introduced, the 'amplitude and phase error terms' (APETs). The reproducing conditions and the criterion to derive the correction function are different from that of the continuous system, differing in the APETs [95]. From the discrete Fourier analysis, we know that APETs are the outcome of the system discretization. For the general case, the APETs decrease as the dilatation parameter increases, but the APETs can not be eliminated from the reproducing process. Another error term arises in the higher order polynomial reproduction (higher than the order of consistency), and can be called reproduction error. This error is introduced by the higher-order derivatives and is proportional to the dilatation. This means that a larger dilatation parameter will cause higher reproduction errors, while the APETs decrease. Therefore, we find a rise and a fall in the error distribution with varying dilatation parameter [95].

This can also be seen in Fig. 4 for example results which are produced with a Bubnov-Galerkin MM and a collocation MM applied for the approximation

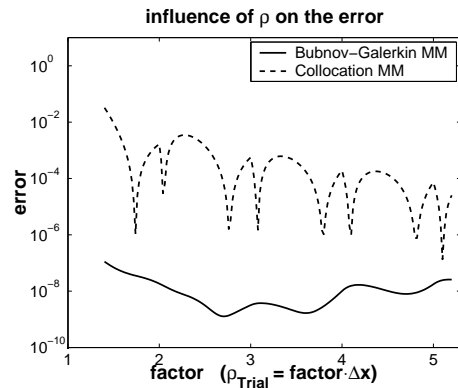


Figure 4: Rise and fall in the error for varying dilatation parameter.

of the one-dimensional advection-diffusion equation. It can clearly be seen that the Bubnov-Galerkin MM gives much better results with small rises and falls in the error plot while the collocation MM shows a strong (and not predictable) dependence on the dilatation parameter and gives worse results in all cases. However, it should already be mentioned here that Bubnov-Galerkin MMs are much more computationally demanding than collocation MMs.

Another important conclusion from the study of the Fourier analysis is that the resolution limit and the resolvable scale of the system are two different issues. The resolution limit, solely determined by the sampling rate, is problem independent and is an unbreakable barrier for discrete systems. On the other hand, the resolvable scale of the system is dictated by the interaction between the system responses, especially its high scale non-physical noise, and the choice of interpolation functions. The Fourier analysis provides the tools to design better interpolation functions which will improve the accuracy of the solution and stretch the resolvable scale toward the resolution limit. [95]

For a global error analysis of the meshfree RKPM interpolants under a global regularity assumption on the particle distribution, see [59]; for a local version —applicable to cases with local particle refinement— see [60].

4.3.1 Hermite RKPM

The name 'Hermite RKPM' stems from the well-known Hermite polynomials used in the FEM for the solution of forth order boundary value problems. In

this approach, presented in [95], the reproducing conditions are enforced in a way that the derivatives can be treated as independent functions:

$$\begin{aligned} u^h(\mathbf{x}) &= \sum_{j=1}^q \left(\int_{\Omega_y} K^{\alpha^j}(\mathbf{x}, \mathbf{y}) D^{\alpha^j} u(\mathbf{y}) d\Omega_y \right) \\ &= \sum_{j=1}^q \left(\mathbf{p}^T(\mathbf{x}) \left[\int_{\Omega_y} w^{(\alpha^j)}(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y}) \mathbf{p}^T(\mathbf{y}) d\Omega_y \right]^{-1} \right. \\ &\quad \left. \int_{\Omega_y} w^{(\alpha^j)}(\mathbf{x} - \mathbf{y}) \mathbf{p}(\mathbf{y}) D^{\alpha^j} u(\mathbf{y}) d\Omega_y \right) \end{aligned}$$

By doing this, we introduce the derivatives of the function as unknowns. Note the similarity to the GMLS presented in subsection 4.2.4.

4.4 Particle Placement

Starting point for any construction of a PU is the distribution of nodes in the domain. Although it is often stated that MMs work with randomly or arbitrary scattered points, the method cannot be expected to give suitable results if several criteria are not fulfilled. For example, Han and Meng introduce for this purpose the concept of (r, p) -regular particle distributions with the essential point that the inverse of the mass matrix is uniformly bounded by a constant [59].

There are methods for producing well-spaced point sets, similar to mesh generators for mesh based methods. Some methods rely on advancing front methods, such as the biting method [90, 89]. Other point set generators are octree based [74] or they use Voronoi diagrams and weighted bubble packing [34]. We do not go into further detail because there are basically the same methods for the distributions of nodes as in mesh-based methods where this is also the first step.

4.5 Weighting Functions

The weighting functions of MLS and RKPM are translated and dilatated. The ability to translate makes elements unnecessary, while dilatation enables refinement [94].

Both meshfree methods —MLS and RKPM— which have been used for the construction of a PU of consistency n , used a weighting (also: kernel or window)

function w which still not has been discussed. As the methods have different origins the motivation for introducing weighting functions are different.

The MLS has its origin in interpolating data and the weighting function has been introduced to obtain a certain locality of the point data due to the compact support. The moving weight function distinguishes the MLS from other least-squares approaches. If all weighting functions are constant, then $u^h(\mathbf{x})$ is a standard non-moving least-squares approximation or regression function for u . In this case the unknown vector $\mathbf{a}(\mathbf{x})$ is a constant vector \mathbf{a} and all unknowns are fully coupled.

The RKPM with its origin in wavelet theory uses the concept of the weighting function already as its starting point: the integral window transform. It can be easily seen that this continuous approximation turns out to be exact, if the weight function $w(\mathbf{x} - \mathbf{y}, \rho)$ equals the Dirac function $\delta(\mathbf{x} - \mathbf{y})$. However, in the discrete version of this RKM, the RKPM, the Delta function has to be used in numerical integration and thus other functions with small supports have to be used [107].

Despite of these rather different viewpoints due to the similarity of the resulting methods, there is also a close similarity in the choice of the weighting functions for MLS and RKPM. The most important characteristics of weight functions are listed in the following.

Lagrangian and Eulerian kernels In MMs the particles often move through the domain with certain velocities. That is, the problem under consideration is given in Lagrangian formulation, rather than in Eulerian form where particles are kept fixed throughout the calculation. Also the weighting (=window) function may be a function of the material or Lagrangian coordinates \mathbf{X} , $w_i(\mathbf{X}) = w(\|\mathbf{X} - \mathbf{X}_i\|, \rho)$, or of the spatial or Eulerian coordinates \mathbf{x} , $w_i(\mathbf{x}) = w(\|\mathbf{x} - \mathbf{x}_i(t)\|, \rho)$. The difference between these two formulations may be seen in Fig. 5a) and b), where particles move due to a prescribed non-divergence-free velocity field. It is obvious that the shape of the support changes with time for the Lagrangian kernel but remains constant for the Eulerian kernel.

An important consequence of the Lagrangian kernel is that neighbours of a particle remain neighbours throughout the simulation. This has large computational benefits, because a neighbour search for the summation of the MLS system of equations has only to be done once at the beginning of a computation. In addition, it has been shown in [14] and [118] that Lagrangian kernels have superior stability properties in collocation MMs, for example they do not suffer from the

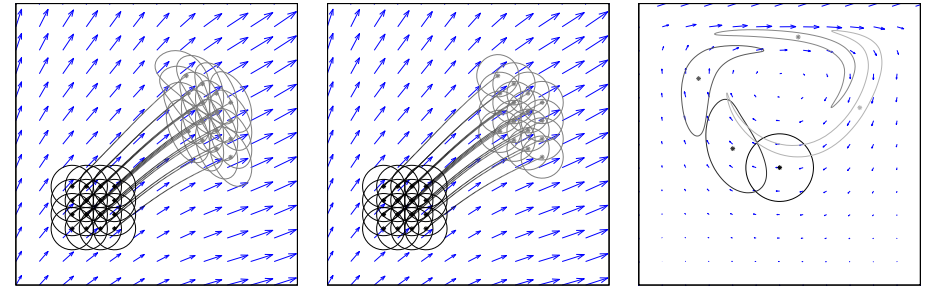


Figure 5: a) and b) compare Lagrangian and Eulerian kernels, c) shows the limited use of Lagrangian kernels. The initial situation at $t = 0$ is plotted in black, grey lines show situations at $t > 0$.

tensile instability (see subsection 5.1). However, the usage of Lagrangian kernels comes at the major disadvantage that it is limited to computations with rather small movements of the particles during the calculation (as is often the case e.g. in structural mechanics) [118]. It can be seen in Fig. 5c) that Lagrangian kernels may not be used in problems of fluid dynamics due to the prohibitive large deformation of the support shape. In this figure a divergence-free flow field resulting from the well-known driven cavity test case has been taken as an example. Clearly, in cases where neighbour relations break naturally —i.e. physically justified— throughout the simulation, Lagrangian kernels seem useless as their property to keep the neighbour relations constant is undesirable.

It is clear that if the problem under consideration is posed in an Eulerian form, then the particle positions are fixed throughout the simulation and the shape of the supports stays constant, i.e. an Eulerian kernel results naturally. For a theoretical analysis of Lagrangian and Eulerian kernels see [14].

In the following, we do not further separate between Lagrangian and Eulerian kernels and write space coordinates rather like \mathbf{x} than \mathbf{X} , without intention to restrict to Eulerian kernels.

Size and shape of the support The *support* $\tilde{\Omega}_i$ of a weight function w_i differs in size and shape, the latter including implicitly the dimension of the PDE problem under consideration. Although any choice of the support shape might be possible, in practice spheres, ellipsoids and parallelepipeds are most frequently used. The size and shape of the support of the weight function is

directly related to the size and support of the resulting shape function, and $\Phi_i(\mathbf{x}) = 0 \forall \{\mathbf{x} | w_i(\mathbf{x}) = 0\}$.

The size of the support is defined by the so-called dilatation parameter or smoothing length ρ . It is critical to solution accuracy and stability and plays a role similar to the element size in the FEM. h -refinement in FEs can be produced in MMs by decreasing the value of the dilatation parameter, thus implying an increase in the density of the particles [98]. Although the dilatation parameter is often chosen to be constant for all points \mathbf{x}_i it can be different for each point and may vary during the calculation. The aim is to reach good solutions — although here it remains unclear how to find optimal smoothing lengths [98] — or to keep the number of particles in the support of each node constant [64]. In both cases, we need to determine time derivatives of ρ , leading to complicated equations. However, Gingold and Monaghan found in [52] that if these terms are omitted, energy is conserved with an error $< 1\%$ or less for large particle numbers N .

Any one-dimensional weighting function $w(\mathbf{x})$ can be used to create a d -dimensional weighting function either of the form $w(\|\mathbf{x}\|)$ in case of spherical supports or by a tensor product $\prod_{i=1}^d w(x_i)$ in case of parallelepipeds.

The intersecting situation of supports $\tilde{\Omega}_i$ is also called *cover*. The cover construction, i.e. the choice of the size (implicitly through the dilatation parameter ρ) and shape of the supports has to fulfill — together with the node distribution — certain conditions in order to ensure the regularity of the $k \times k$ system of equations (moment matrix) which arises in the MLS/RKPM procedure, see subsection 4.6. The aspect of an automatic cover construction for a given point set is worked out in [53]. However in practice, instead of using certain algorithms for the definition of the cover, it is often constructed manually in a straightforward “intuitive” way.

Functional form of the weighting function The third important characteristic of weight functions is their *functional form*. In general, w is chosen to be a member of a sequence of functions which approximates a δ function [52], in accordance with the RKPM point of view. There exist infinitely many possible choices [59, 108] but typically, exponential (Gaussian) functions or spline functions of different orders are used. The functional form has some effect on the convergence of an approximation and is difficult to predict [98]. The weighting function should be continuous and positive in its support.

An important consequence of the choice of the functional form is the continuity (smoothness) of the approximation. The smoothness of the resulting shape

function is directly related to the smoothness of the weight function. Provided that the basis \mathbf{p} is also at least as continuous as the weighting function w , then if w is continuous together with its first l derivatives, i.e. $w \in C^l(\Omega)$, the interpolation is also continuous together with its first l derivatives. More general, if $\mathbf{p} \in C^m(\Omega)$ and $w \in C^l(\Omega)$, then the shape function $\Phi \in C^{\min(l,m)}(\Omega)$, see e.g. [40] for a proof.

We give some examples of frequently used weighting functions:

$$\begin{aligned} 3^{rd} \text{ order spline : } w(q) \in C^2 &= \begin{cases} \frac{2}{3} - 4q^2 + 4q^3 & q \leq \frac{1}{2} \\ \frac{4}{3} - 4q + 4q^2 - \frac{4}{3}q^3 & \frac{1}{2} < q \leq 1 \\ 0 & q > 1 \end{cases} \\ 4^{th} \text{ order spline : } w(q) \in C^2 &= \begin{cases} 1 - 6q^2 + 8q^3 - 3q^4 & q \leq 1 \\ 0 & q > 1 \end{cases} \\ 2k^{th} \text{ order spline : } w(q) \in C^{k-1} &= \begin{cases} (1 - q^2)^k & q \leq 1 \\ 0 & q > 1 \end{cases} \\ \text{singular : } w(q) \in C^0 &= \begin{cases} q^{-k} - 1 & q \leq 1 \\ 0 & q > 1 \end{cases} \\ \text{exponential 1 : } w(q) \in C^{-1} &= \begin{cases} e^{-(q/c)^{2k}} & q \leq 1 \\ 0 & q > 1 \end{cases} \\ \text{exponential 2 : } w(q) \in C^0 &= \begin{cases} \frac{e^{-(q/c)^{2k}} - e^{-(1/c)^{2k}}}{1 - e^{-(1/c)^{2k}}} & q \leq 1 \\ 0 & q > 1 \end{cases} \\ \text{exponential 3 : } w(q) \in C^\infty &= \begin{cases} e^{1/(q^2-1)} & q \leq 1 \\ 0 & q > 1 \end{cases} \end{aligned}$$

where $q = \frac{\|\mathbf{x} - \mathbf{x}_i\|}{\rho}$. The difference between the two exponential weighting functions is that version 1 is not zero at the boundary of the support, because $w(1) = e^{-(1/c)^{2k}} \neq 0$, thus it is not continuous (C^{-1}). Version 2 fixes this lack as it shifts the weighting function by subtracting $e^{-(1/c)^{2k}}$ to have $w(1) = 0$ and then divides through $1 - e^{-(1/c)^{2k}}$ to still have $w(0) = 1$. In Fig. 6 the 3^{rd} and 4^{th} order spline weighting functions are shown together with the Gaussian weighting function (version 2) for different values of c and $k = 1$.

As shown in subsection 4.2, the MLS approximant $u^h = Gu$, does in general not interpolate (“pass through”) the data, which might be disadvantageous. Already Lancaster and Salkauskas pointed out in [82] that the interpolating property of the shape functions can be recovered by using singular weighting functions at *all* nodes. Then we can obtain Kronecker-Delta property of the interpolating functions.

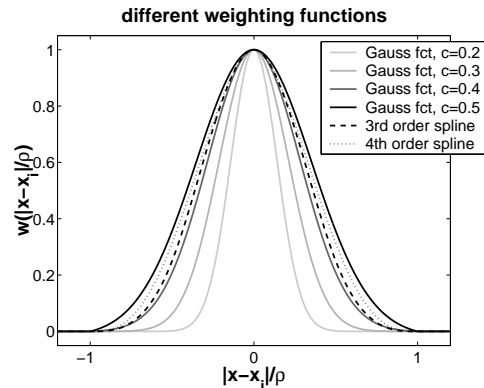


Figure 6: Exponential and spline weighting functions.

It should be mentioned that the support size (defined by ρ) and shape as well as the functional form of the weighting function are free values. It is impossible to choose these values in a general, optimal way suited for arbitrary problems under consideration. However, one may take advantage of these free values in obtaining certain properties of the approximation method. For example, in [71] Jin *et al.* modify the weighting in the framework of meshfree collocation methods to enable the fulfillment of the so-called “positivity conditions” (which also arise in a finite difference context). Atluri *et al.* modify either the functional form of the weighting function or shift the support in upwind direction to obtain stabilizing effects in a Galerkin setting of advection-dominated problems [4]. On the other hand, it may also be a major problem in certain cases to have free values to define the characteristics of the weighting function without knowing how to choose them. For example, intuitive *ad hoc* approaches such as keeping the ratio between the particle density and smoothing length $\frac{\rho}{h}$ constant when changing the particle number locally seems straightforward, however, in the context of standard SPH this may not even converge [15]. Or an improper choice of a parameter may result in instability of the numerical solution (small changes of improperly selected parameter evoke large fluctuations in the solutions), see e.g. [5]. In these situations it is obviously not desirable to have these free values. Despite of these considerations, it should be added that it is in practice often not difficult to choose the free parameters and obtain satisfactory results.

Some general statements about weight functions are cited in the following:

Every interpolation scheme is in fact related to a filter [95]. The frequency

content of the solution is limited to the frequency spectrum of the scheme. If *a priori* knowledge on the character of a set of differential equations is available, an efficient interpolation scheme can be developed by designing special window functions (filters) [95]. A large window (in the function domain) filters out the fine scales (small wave numbers or high frequencies) and the frequency band becomes small and oscillation in the solution may occur [94, 97]. In contrast, a small window may introduce aliasing but will cover wider scale/frequency bands [94, 97]. If it were possible to reproduce the Dirac delta function, all frequencies of a function would be reproduced [97].

In the space domain Gaussian functions have infinite support, while splines are compactly supported. However, in the Fourier transform domain an ideal low-pass region exists [94]. Here, the Gaussian window decays rapidly outside this ideal filter region while splines show side-lobes towards infinity. These side-lobes will introduce aliasing in the reconstruction procedure. The side-lobes of the spline family in the frequency domain decrease as the polynomial order of the window function increases. This decrease in side-lobes will prevent additional high frequency aliasing from being introduced into the system response [94]. One may thus follow that the quickly decaying Gaussian window function is recommendable to reduce the error [95].

4.6 Solving the $k \times k$ System of Equations

In both methods, MLS and RKPM, in order to evaluate the n -th order consistent shape functions at a certain point \mathbf{x} a $k \times k$ matrix, the moment matrix $\mathbf{M}(\mathbf{x})$, must be inverted, i.e. a system of equations must be solved. The parameter k , which defines the size of this system, equals the number of components in the intrinsic basis $\mathbf{p}(\mathbf{x})$, and thus depends on the dimension of the problem d and the consistency order n , see subsection 2.4. In order to evaluate the integral expressions of the weak form of a PDE problem, a large number of integration points \mathbf{x}_Q has to be introduced. At each of these points the $k \times k$ system has to be built and to be solved.

The need to build up and invert the moment matrix at a large number of points is the major drawback of the MMs, because of the computational cost and the possibility that the matrix inversion fails (in contrast to the FEM). The computational cost consists in evaluating summation expressions including a neighbour search and in matrix inversion itself. Furthermore, the computation of the derivatives of the shape functions involves large number of (small) matrix-matrix and matrix-vector multiplications, see Eqs. 4.2 and 4.3.

A sum over all particles has to be evaluated for the construction of the moment matrix $\mathbf{M}(\mathbf{x}) = \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i)$ and the right hand side $\mathbf{B}(\mathbf{x}) = \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i)$. This requires the identification of the particles' neighbours, i.e. the detection of particles with $w(\mathbf{x} - \mathbf{x}_j, \rho) \neq 0$. This may be called connectivity computation; note that in mesh-based methods, the mesh defines the connectivity *a priori*. In MMs the connectivity is determined at run-time for each point at which the shape functions need to be evaluated. This important step can dominate the total CPU time for large point numbers, especially if sequential searches, which are of $O(N)$ complexity, are used for each evaluation point. Therefore, one should always try to use search techniques which employ localization, since such techniques can perform the search at a given point in an optimal time $O(\log N)$ [80].

The moment matrix $\mathbf{M}(\mathbf{x})$ is symmetric and under certain conditions it is expected to be positive-definite. The matrix inversion is usually done via a factorization by the pivoting LU, QR factorization or singular value decomposition (the latter two are indicated for ill-conditioned matrices) [80]. A factorization of $\mathbf{M}(\mathbf{x})$ can be reused for the calculation of the shape functions derivatives, so that this involves only little extra effort [16].

Despite the computational burden associated with the construction and inversion of the matrix, the inversion can even fail if $\mathbf{M}(\mathbf{x})$ becomes singular (the rank of $\mathbf{M}(\mathbf{x})$ becomes smaller than k) or “nearly” singular, hence ill-conditioned. Conditions on the particle distribution (subsection 4.4) and cover (subsection 4.5) in order to ensure the regularity of the mass matrix are:

- For every point $\bar{\mathbf{x}} \in \Omega$ there exists a ball $B(\bar{\mathbf{x}}) = \{\mathbf{x} \mid \|\mathbf{x} - \bar{\mathbf{x}}\| \leq c\}$ in which the number of particles \bar{N} satisfies the condition $0 < \bar{N}_{\min} \leq \bar{N} \leq \bar{N}_{\max} < \infty$ where \bar{N}_{\min} and \bar{N}_{\max} are *a priori* numbers [99].
- Each particle i has a corresponding support $\tilde{\Omega}_i$ (where $w(\mathbf{x} - \mathbf{x}_i) \neq 0$). The union of all supports covers the whole domain, e.g. $\Omega \subseteq \bigcup_{i=1}^N \tilde{\Omega}_i$ [99].
- Every point $\mathbf{x} \in \Omega$ must lie in the area of influence of at least $k = \dim(\mathbf{M})$ particles [65], hence:

$$\text{card}\{\mathbf{x}_j \mid w(\mathbf{x} - \mathbf{x}_j) \neq 0 \ \forall j \in \{1, 2, \dots, N\}\} \geq k = \dim(\mathbf{M}).$$

- The particle distribution must be non-degenerate [65, 99]. For example, $d + 1$ particles are needed for the construction of a PU of *first* order and they must describe a non-degenerate d -simplex: In two dimensions \mathbf{x} must belong to at least three supports of particles which are not aligned, and

in three dimensions \mathbf{x} must belong to at least four supports of particles which are not coplanar.

A robust algorithm should always check the success of the matrix inversions [80]. There are two possible checks. The first one consists in estimating the condition number of $\mathbf{M}(\mathbf{x})$ and the second by ensuring the final accuracy of the shape functions by checking the fulfillment of the consistency conditions (Eq. 2.2), possibly including the derivative consistency conditions (Eq. 2.3). Thus it is checked if really a PU of the desired order has been obtained. If a certain mismatch is exceeded an error exit can be made. Of course, satisfying the conditions for regular matrices \mathbf{M} does not ensure the regularity (and consequently solvability) of the total stiffness matrix [65].

In the MLS and RKPM we use the basis of monomials $\mathbf{p}(\mathbf{x})$ which can easily lead to ill-conditioned moment matrices. However, it was already mentioned in 4.2 that any linear combination of the basis functions will lead to the same shape functions, and according to this, translated and scaled bases can be used leading to a better conditioning of the moment matrices. In general, with the translation to x_j and scaling with ρ the matrix has a lower condition number [65]. Instead of $\Phi^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) [\mathbf{M}(\mathbf{x})]^{-1} \mathbf{B}(\mathbf{x})$ one may write $\Phi^T(\mathbf{x}) = \mathbf{p}^T\left(\frac{\mathbf{x} - \mathbf{x}_i}{\rho}\right) [\mathbf{M}(\mathbf{x})]^{-1} \mathbf{B}(\mathbf{x})$, with $\mathbf{M}(\mathbf{x}) = \sum_{i=1}^N \mathbf{p}\left(\frac{\mathbf{x} - \mathbf{x}_i}{\rho}\right) \mathbf{p}^T\left(\frac{\mathbf{x} - \mathbf{x}_i}{\rho}\right) w\left(\frac{\mathbf{x} - \mathbf{x}_i}{\rho}\right)$ and $\mathbf{B}(\mathbf{x}) = \sum_{i=1}^N w\left(\frac{\mathbf{x} - \mathbf{x}_i}{\rho}\right) \mathbf{p}(0)$. When the dilatation parameter ρ varies at each particle another definition of the shape functions is recommended: For $\mathbf{B}(\mathbf{x})$ the weighting function is scaled with $w\left(\frac{\mathbf{x} - \mathbf{x}_j}{\rho_j}\right)$ instead of $w\left(\frac{\mathbf{x} - \mathbf{x}_j}{\rho}\right)$, where ρ_j is the dilatation parameter associated with particle j , and ρ is the constant average value of the particles' dilatation parameters [65].

4.7 Summary and Comments

In this section we showed in detail how partitions of unities can be constructed, in both meshfree and mesh-based ways.

We found that the two starting points for MLS and RKPM are very different but lead to almost identical results for the PU functions. In the MLS one deals with discrete sums from the beginning while the RKM starts with continuous integrals which later on lead to discrete sums when these integrals are evaluated numerically. The discrete versions of MLS and RKPM are in general identical, however, this is not necessarily the case and depends on the choice of the particle volumes ΔV_i in the RKPM.

It is important to mention that in [99, 85] a continuous background for the MLS was deduced, called the MLSRK. The MLSRK is the continuous counterpart of the MLS and is somehow the last missing link to show equivalence of the RKPM and MLS concepts. In this generalization of the discrete MLS we obtain the same integral expressions as for the RKPM with the freedom to choose adequate integration weights. This again underlines the similarity of MLS and RKPM, thus whenever we say MLS functions in this paper one could analogously put in RKPM functions. We conclude that for the choice or the designing of a MM for a certain purpose the question whether the MLS or RKPM should be used as the underlying principle for the construction of the PU function is of less importance.

In subsection 4.5 different weighting functions —also known as window functions— have been presented. The spline functions seem to be used most often in practice, they have the advantage that no additional free values have to be adjusted as for example for the Gaussian weighting function. The dependence of the accuracy on the dilatation parameter ρ for the definition of the support size has been pointed out, no optimal universal value may be specified. In subsection 4.6 it was further stressed that the dilatation parameter controls the regularity of the mass matrix and thereby the stability of the MLS and RKPM procedures.

Some aspects of Eulerian and Lagrangian formulations have been mentioned in 4.5; the same aspects have to be discussed for MMs than for any other method. In the Eulerian formulation the point positions are constant throughout the whole calculation and no update of the mesh is necessary. In case of moving boundaries it may be difficult to keep the pure Eulerian viewpoint, e.g. in free-surface flow or large deformation analysis. In Eulerian formulations advection terms often arise in the model equations of a problem. These terms are related with major problems in the numerical solution of these equations and usually require some kind of stabilization; for a detailed discussion of stabilization for MMs see e.g. [47, 48]. Imposition of BCs is often rather easy for Eulerian methods as one can impose them at *fixed* boundary nodes.

Lagrangian methods require a permanent update of the approximation points involved. Mesh-based methods are often difficult to use in this case due to the expansive process of update conforming non-structured FE meshes, however, this aspect does not arise for MMs. The advantage for model equations in Lagrangian formulation is that no advection terms arise and consequently no stabilization of these terms is necessary and accuracy is often better than in Eulerian methods. The treatment of BCs in Lagrangian methods is rather complicated at least in the presence of boundaries where points (e.g. matter) enter or

leave the domain. To insure the invertibility of the $k \times k$ system of equations it is necessary to control the *moving* point positions and dilatation parameters; also the “variable rank MLS” (subsection 5.1) may be used. Taking the Lagrangian formulation one may still decide whether an Eulerian or Lagrangian *kernel* is preferred, i.e. whether the support size and shape is constant or changes, see 4.5. The usage of a (deforming) Lagrangian kernel is limited to small deformations. Eulerian kernels move unchanged with the corresponding particles and may be used for arbitrary deformations —however, the reduced stability may be a problem.

5 Specific Meshfree Methods

In the previous chapter it is explained how a partition of unity with n -th order consistency can be constructed, either mesh-based or meshfree. For this purpose a basis $\mathbf{p}(\mathbf{x})$ was introduced, which is called *intrinsic* basis.

The next step is to define the approximation. Most of the MMs to be discussed below simply use an approximation of the form

$$u^h(\mathbf{x}) = \sum_{i=1}^N \Phi_i(\mathbf{x}) u_i = \mathbf{\Phi}^T(\mathbf{x}) \mathbf{u}$$

so that the PU functions $\{\Phi_i(\mathbf{x})\}$ are directly taken as trial functions in the approximation. However, there exists also the possibility to use an *extrinsic* basis in the approximation which can serve the following purposes: Either to increase the order of consistency of the approximation or to include *a priori* knowledge of the exact solution of the PDE into the approximation. This will be discussed in more detail in subsections 5.7 and 5.8.

Summarizing this, it can be concluded that the approximation can either be done with usage of an intrinsic basis only or with an additional extrinsic basis. After defining the approximation $u^h(\mathbf{x})$ and inserting this for $u(\mathbf{x})$ in the weak form of the method of weighted residuals, test functions have to be defined. Then the integral expressions of the weak form can be evaluated and the system matrix and right hand side can be constructed.

With definition of the partition of unity, the approximation and the test functions the herein considered methods can be clearly identified as shown in Fig. 1 on page 16.

Before discussing the individual MMs in the following subsections a few general remarks shall be made with respect to collocation and Galerkin MMs;

a number of further details may then be found in the separate subsections. Collocation methods result from choosing the Dirac delta function as a test function. Then the weak form of a PDE reduces to the strong form and the integral expressions vanish. For any other choice of the test functions a Galerkin method results. If the test functions are chosen identically to the trial function a Bubnov-Galerkin method results, in any other case a Petrov-Galerkin method follows.

The advantage of collocation methods is their efficiency in constructing the final system of equations which is due to the fact that integration is not required and shape functions are evaluated at nodal positions only. However, the accuracy and robustness of collocation approaches are weak points especially if the approximation is based on a set of randomly scattered points. Collocation methods may lead to large numerical errors in these cases and involve numerical stability problems. An interesting discussion of these aspects may be found in [71] where certain conditions are proposed which are often not met by standard collocation MMs. Furthermore, the boundary conditions are an issue for collocation methods. The definition and location of the boundary surface may not be easy, e.g. for free surface flow etc., and methods of applying boundary conditions are not always straightforward (natural boundary conditions).

The advantages and disadvantages of Galerkin methods are often vice versa. Boundary conditions are treated easily, accuracy and robustness are in general no problems. The weak point of these methods is the necessity of a sufficiently accurate integration which requires a large number of integration points and shape function evaluations; this issue is discussed separately in subsection 6.2.

5.1 Smoothed Particle Hydrodynamics (SPH, CSPH, MLSPH)

The SPH method was introduced in 1977 by Lucy in [104], Monaghan worked this further out in [107] by using the notion of kernel approximations. The SPH is a Lagrangian particle method and is in general a representative of a strong form collocation approach. SPH is the first and simplest of the MMs, it is easy to implement and reasonably robust [108]. SPH was motivated by ideas from statistical theory and from Monte Carlo integration and was first used for astronomical problems [52]. The name SPH stems from the smoothing character of the particles' point properties to the kernel function, thus leading to a continuous field.

First we consider the statistical viewpoint, which is the origin of the SPH,

by regarding the approximation of the density ρ in fluid problems. Gingold and Monaghan claim in [52]:

The density ρ for particles of equal mass is proportional to the number of particles per volume. The same can be considered from a statistical point of view: the probability that a particle is found in the volume element ΔV is proportional to $\rho \Delta V$. If the statistical point of view is carried over to the system of fluid elements, the density can be defined in the same way. We regard the positions of the fluid elements as a random sample from a probability density proportional to the mass density. The estimation of the density is then equivalent to estimating a probability density for a sample. Known statistical methods based on smoothing kernels can be used for this purpose. The statistical estimation of density by smoothing kernels can be interpreted as the replacement of each particle by a smoothed-out density (hence we call it smoothed particle hydrodynamics SPH).

Shortly, the density can be considered either as proportional to the average number of particles per unit volume, or as proportional to probability density of finding a particle in a given volume element. With the latter interpretation we consider the estimate of the true density ρ :

$$\rho^h(\mathbf{x}) = \int_{\Omega_{\mathbf{y}}} w(\mathbf{x} - \mathbf{y}) \rho(\mathbf{y}) d\Omega_{\mathbf{y}},$$

with $\int_{\Omega_{\mathbf{y}}} w(\mathbf{x} - \mathbf{y}) d\Omega_{\mathbf{y}} = 1$, motivated from statistics. Note that only here ρ stands for the density and not for the dilatation parameter. Since $\rho(\mathbf{y})$ is unknown, the above expression cannot be evaluated directly, but if we have a set of N randomly distributed points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ according to ρ , the integral can be evaluated by the Monte Carlo method as

$$\rho_N^h(r) = \frac{M}{N} \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i),$$

with M being the total mass $M = \int_{\Omega_{\mathbf{x}}} \rho^h(\mathbf{x}) d\Omega_{\mathbf{x}}$.

Now we leave this special consideration for the approximation of the density in fluid problems and generalize the above for the approximation of an arbitrary

PDE:

$$\begin{aligned} u^h(\mathbf{x}) &= \int_{\Omega_{\mathbf{y}}} w(\mathbf{x} - \mathbf{y}) u(\mathbf{y}) d\Omega_{\mathbf{y}} \\ &= \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \Delta V_i u(\mathbf{x}_i). \end{aligned}$$

The SPH was introduced for *unbounded* problems in astrophysics [52, 108] and applying it to bounded cases leads to major problems. This is due to its failure to meet the reproducing conditions of even 0-th order near the boundaries. This can be easily shown by a Taylor series analysis. For 0-th order consistency we need

$$\begin{aligned} u^h(\mathbf{x}) &= \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \Delta V_i u_i \\ &= \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \Delta V_i \left(u(\mathbf{x}) + \sum_{|\alpha|=1}^{\infty} \frac{(\mathbf{x}_i - \mathbf{x})^\alpha}{|\alpha|!} D^\alpha u(\mathbf{x}) \right) \\ &= \sum_{i=1}^N w(\mathbf{x} - \mathbf{x}_i) \Delta V_i u(\mathbf{x}) + \text{error}. \end{aligned}$$

Thus, the kernel sum $\sum_{i=1}^N w(\mathbf{x}) \Delta V_i$ must equal 1 in the whole domain to fulfill this equation, i.e. to have an approximation of 0-th order. It is recalled that in case of RKPM consistency could be reached due to a correction function, which obviously misses in SPH (ΔV_i stands for integration weights and not for a correction term). Thus consistency cannot be reached at boundaries, where $\sum_{i=1}^N w(\mathbf{x}) \Delta V_i \neq 1$, which can be easily seen from the dotted line in Fig. 7. The shape functions of SPH are $\Phi_i(\mathbf{x}) = w(\mathbf{x}) \Delta V_i$, thus in this case of regularly distributed nodes in one dimension, all inner shape functions are identical, due to $\Delta V_i = (x_{i+1} - x_{i-1})/2 = \text{const}$ and only the nodes on the boundaries have different shape functions, due to $\Delta V_1 = \Delta V_N = \Delta V_i/2$ (one may also choose $\Delta V_i = 1$ for all nodes).

The lack of consistency near boundaries leads to a solution deterioration near the domain boundaries, also called “spurious boundary effects” [25]. SPH also shows the so-called “tension instability”, first identified by Swegle *et al.* [127], which results from the interaction of the kernel with the constitutive relation. It is independent of (artificial) viscosity effects and time integration algorithms. It has been shown by Dilts in [38] that the tension instability is directly related

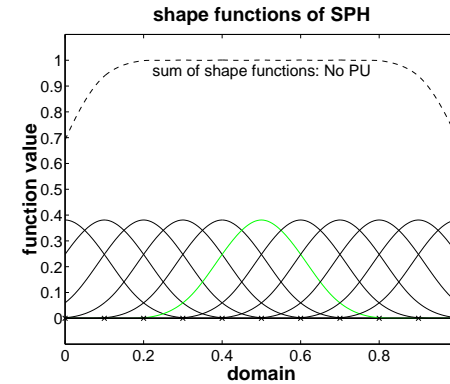


Figure 7: Shape functions constructed with the SPH. They do not build a PU, in particular not near the boundary.

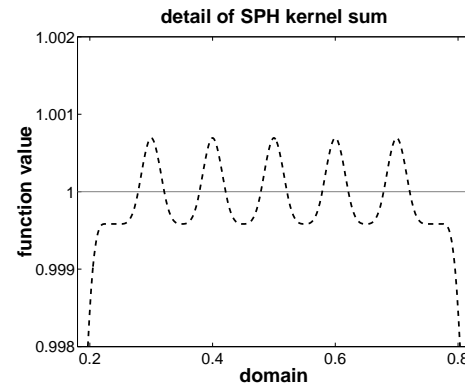


Figure 8: Detail of the kernel sum of the SPH shape functions of Fig. 7. These small oscillations give rise to instabilities in SPH.

to the appearance of oscillations in the kernel sums (which in SPH are *not* exactly 1, as SPH shape functions do *not* form a partition of unity). This can be seen in Fig. 8. If these oscillations can be eliminated or in other words: if consistency can be reached, the tension instability vanishes [38]. It is also important to note that the tension instability is a consequence of using Eulerian kernels (see subsection 4.5) in a Lagrangian collocation scheme; it does not occur for Lagrangian kernels [14].

Another instability in SPH (and other collocation MMs) results from the rank deficiency of the discrete divergence operator [14] and occurs for Eulerian as well as for Lagrangian kernels.

There are several ideas to stabilize these problems. One approach is to use so-called stress points [44, 119]. The name is due to the fact that stresses are calculated at these points by the constitutive equation in terms of the particle velocities [14]. Its extension to multi-dimensional cases is not easy as stress points must be placed carefully [14, 119].

We summarize the idea of the SPH as follows: In SPH a computational domain is initially replaced by discrete points, which are known as particles. They represent any field quantities in terms of its values and move with their own (fluid) velocities, carrying all necessary physical information. These particles can be considered as moving interpolation points. In order to move the particles correctly during a time step it is necessary to construct the forces which an element of fluid would experience [108]. These forces must be calculated from the information carried by the particles. The use of interpolation kernel allows smoothed approximations to the physical properties of the domain to be calculated from the particle information. This can be interpreted as smoothing the discrete properties of the points over a finite region of space and hence led to the name SPH [81].

It has already been mentioned above that the treatment of boundaries is one major drawback in the SPH, which has been pointed out in a many references, see e.g. [87]. In fact, it differs from the other MMs. There is no systematic way to handle neither rigid nor moving boundaries [98]. According to [108], rigid walls have been simulated using (a) forces with a length scale h (this mimics the physics behind the boundary condition), (b) perfect reflection and (c) a layer of fixed particles. The fixed particles in the latter approach are often called “ghost particles”, see e.g. [119] where boundary conditions in SPH have been intensively discussed. Natural boundary conditions are also a major problem in SPH and collocation methods in general [14].

It should also be mentioned, concerning the h -adaptivity of the SPH, that Belytschko *et al.* claim in [15] that SPH does not necessarily converge if the

size of the smoothing length is kept proportional to the distance between nodes $\rho/h = \text{const}$ —that is, a standard refinement procedure of adding particle and simultaneously decreasing the support size may fail. In fact, convergence proofs for the SPH assume certain more demanding relationships between nodal spacing and support size [15]. As a consequence the sparsity of the equations decreases drastically leading to a severe drop in computational efficiency.

Improvements of the standard SPH method are still an active research area and there exists a number of other proposed correction ideas for the SPH addressing tensile instability, boundary conditions and consistency; see e.g. [87], [15] and [119] for an overview and further references. The approaches to fix certain lacks of the SPH differ in their complexity and computational effort. We only describe briefly two ideas of correcting the SPH, both approaches obtain a certain consistency of the SPH shape functions. One may also interpret the RKPM shape functions (subsection 4.3) in a collocation setting, see e.g. [1], as a corrected SPH with the ability to reproduce polynomials exactly. The Finite Point Method (FPM), introduced by Oñate *et al.* in [115], is also a consistent collocation method which is based on fixed (Eulerian) particles in contrast to the moving (Lagrangian) particles of the SPH.

Corrected Smoothed Particle Hydrodynamics (CSPH)

The CSPH is based on a number of correction terms to the standard SPH with the aim to achieve first order consistent solutions without spurious modes or mechanisms [22, 81].

Instead of w a corrected kernel $\hat{w}_i(\mathbf{x}) = w_i(\mathbf{x})\alpha(\mathbf{x})[1 + \beta(\mathbf{x})(\mathbf{x} - \mathbf{x}_i)]$ is introduced, where α and β are evaluated by enforcing consistency conditions [22]. The resulting method is called CSPH. The correction largely improves the accuracy near or on the boundaries of the problem domain. Next, the discrepancies that result from point integration are being addressed by introducing an integral correction vector γ . This enables the integration corrected CSPH to pass the linear patch test [22].

A last correction term is introduced to stabilize the point-wise integration of the SPH and thus prevent the emergence of spurious modes or artificial mechanisms in the solution. The stabilization technique of the CSPH is based on Finite Increment Calculus [81] or least-squares approaches [22]. The cause of spatial instabilities due to point-based integration is described in detail in [22]:

The point-based integration used in the CSPH method relies on the evaluation of function derivatives at the same point where the function values are sought. It is well known in the finite difference

literature that this can lead to spurious modes for which the derivative is zero at all points considered. The simplest example of these problem is encountered when the 1D central difference formula for the first derivative is used as $u'_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x}$. Clearly, there are two solution patterns or “modes” for which the above formula gives zero derivatives at all points. The first is obtained when the function is constant, e.g. $u = 1 \Rightarrow u_a = 1$ for all a . Then the result is correct as u' shall be zero here. The second possibility, however, emerges when $u_a = (-1)^a$. This is clearly an invalid or spurious mode. They will not contribute towards the point integrated variational principle and are consequently free to grow unchecked and possibly dominate and therefore invalidate the solution obtained. It is easy to show that spurious modes can also be found in the CSPH equations that are used for the evaluation of the derivative.

Moving Least-Squares Particle Hydrodynamics (MLSPH)

Many of the undesirable features of the SPH occur due to the lack of even zeroth-order consistency. One can easily illustrate why the SPH fails to be even zeroth order consistent (see above). Only if we satisfy $\sum_i w(\mathbf{x}) \Delta V_i = 1$, we would have zeroth order consistency and thus, a constant function could be interpolated exactly. Initially, this can be reached by either inverting a large matrix (which can lead to negative masses) or by arranging the particles initially in a certain way, often a cubic lattice. But as the particles start moving the equation will not be satisfied any longer.

The use of MLS-interpolants fixes the lack of consistency [38]. The sum of all kernel functions then exactly forms a partition of unity. The following relation between SPH and MLSPH can be shown:

$$\Delta V_i w(\mathbf{x}) \rightarrow \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p} w(\mathbf{x}),$$

where \mathbf{M} is the MLS moment matrix and \mathbf{p} the MLS basis vector. This leads to an interpretation of the factor $\mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}$ as a space-dependent volume ΔV_I associated with a particle. It is rather a numerical volume than a physical or geometric volume [38].

In the MLSPH the idea of a “variable rank MLS” can be used, introduced in [38]: In divergent flow situations, due to the movement of particles, the number of particles in each other trial function supports decreases. The idea is now to reduce the consistency order of a particle I which does not have enough neighbours to invert the MLS- $k \times k$ -matrix. For this particle we iterate down

the consistency order until an invertible MLS-matrix is found (a zeroth-order MLS matrix is a 1×1 -matrix which is always invertible). This, however, leads to different consistency orders which in MLS necessarily introduces discontinuities, thus the shape functions are not smooth.

5.2 Diffuse Element Method (DEM)

The DEM was introduced by Nayroles *et al.* in [111]. Although they did not note this fact the interpolants they used in their method were introduced and studied by Lancaster and Salkauskas and others and called MLS interpolants in curve and surface fitting [18, 103]. Nayroles *et al.* had a different viewpoint of their method as a generalization of the FEM.

In [111] they consider the FEM as a special case of a least-squares procedure:

The FEM uses piecewise approximations of unknown functions which are written on a given element e as $u^e(x) = \sum_{j=1}^m p_j(x) a_j^e$ where p is a vector of m independent functions and a^e is a vector of m approximation parameters, constant on element e . Expressing that the values u_i at the n^e nodes x_i of element e provides a set of linear

$$\text{relations between } a_j^e \text{ and } u_i: \{u_i\} = \begin{bmatrix} \dots \\ p_j(x_i) \\ \dots \end{bmatrix} \{a^e\} = [P_n] \{a^e\}. \text{ If}$$

n^e is equal to m , the matrix $[P_n]$ may in general be inverted, leading to the standard shape functions $N_i(x)$: $u(x) = p_j(x) [P_n]^{-1} \{u_i\} = N_i(x) \{u_i\}$. This interpolation procedure may also be seen as minimizing the following expression with respect to a^e for a given element e : $J^e(a^e) = \sum_{i=1}^{n^e} w_i^e (u_i - u^e(x_i))^2$, where $w_i^e = 1$ if node i belongs to the element e and $w_i^e = 0$ otherwise.

The basic idea of the diffuse approximation is to replace the FEM interpolation, valid on an element, by a local weighted least-squares fitting, valid in a small neighbourhood of a point \bar{x} , and based on a variable number of nodes. The approximation function is made smooth by replacing the discontinuous w_i^e coefficients by continuous weighting functions $w^{\bar{x}}(x)$ evaluated at x_i . The vanishing of these weighting functions at a certain distance from the point \bar{x} preserves the local character of the approximation. Around a point \bar{x} , the function $u^{\bar{x}}(x)$ is locally approximated by an expression equivalent

to the one above: $u^{\bar{x}}(x) = \sum_{j=1}^n p_j(x) a_j^{\bar{x}}$. The coefficients $a_j^{\bar{x}}$ corresponding to the point \bar{x} are obtained by minimizing the following expression: $J^{\bar{x}}(a^{\bar{x}}) = \sum_{i=1}^n w_i^{\bar{x}} (u_i - u^{\bar{x}}(x_i))^2$.

It can be followed that each evaluation point of the DEM may be considered as a particular kind of finite element with only one integration point, a number of nodes varying from point to point and a diffuse domain of influence [111]. It can be seen that the classical FEM is just a special case of the DEM, where the weight function is constant over selected subdomains [111].

The DEM approximation can directly be obtained by the MLS approximation, although this was not realized by Nayroles [76]. Although the shape functions of the DEM are identical to the MLS shape functions, Nayroles *et al.* made a number of simplifications:

- They estimate the derivative of a function u by differentiating only $\mathbf{p}(\mathbf{x})$ with respect to \mathbf{x} and considering $\mathbf{a}(\mathbf{x})$ as a constant [111]. Thus e.g. for the first derivative follows $u_{,j}^h(\mathbf{x}) = \sum \mathbf{p}_{,j}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}) = \mathbf{p}_{,j}^T(\mathbf{x}) \mathbf{M}^{-1}(\mathbf{x}) \mathbf{B}(\mathbf{x}) \mathbf{u}$, assuming that $\mathbf{a}(\mathbf{x})$ is constant, hence $\mathbf{a}_{,j}(\mathbf{x}) = 0$. This incorrectness introduces problems and turns out to be the major difference to the EFG where the derivatives are obtained “correctly”.
- They use a very low quadrature rule for integration [103]. Nayroles claims that it is easy to introduce the DEM into existing FEM codes, by using the existing integration points as the diffuse elements and in some cases they even use less integration points than in FEM [111]. However, the opposite is true and in MMs in general we need much more integration points for accurate results.
- They did not enforce EBCs accurately [103].

As a consequence the DEM does not pass the patch test, which is analogous to a fail in consistency [103].

Petrov-Galerkin Diffuse Element Method (PG DEM)

The PG DEM is a modified version of the DEM which passes the patch test. It was introduced by Krongauz and Belytschko in [76] rather to show the reason why DEM does not pass the patch test than introducing a new method in practice. This method is based on a Petrov-Galerkin formulation where test functions are required to meet different conditions than trial functions.

Krongauz and Belytschko discovered an interesting property of DEM approximations [76]:

The derivative approximations are [first order] consistent but not integrable. Thus, the failure to pass the patch test comes from the violation of the following condition: the test functions derivatives must be integrable and the test functions must vanish on a contour enclosed by the domain of the PDE. These conditions are not met by the DEM approximation with simplified derivatives.

Thus, the DEM shape function derivatives $\Phi_{,j} = \mathbf{p}_{,j}^T(x) \mathbf{M}^{-1}(x) \mathbf{B}(x)$ do satisfy the linear consistency requirements although the derivative has been simplified by assuming the coefficients $\mathbf{a}(\mathbf{x})$ to be constant [76]. However, it leads to derivatives which are not integrable and thus, the DEM derivatives are in a sense pseudo-derivatives [76]. It has been proven that shape function derivatives which are not integrable will not pass the patch test.

To fix this lack of integrability, a special case of the Petrov-Galerkin method is introduced. Here, the trial functions Φ_I are not defined, but instead the two derivatives of u are approximated independently. The approximation is $u_{,x}^h = \mathbf{p}_{,x}^T(x) \mathbf{M}^{-1}(x) \mathbf{B}(x) \mathbf{u}$ and $u_{,y}^h = \mathbf{p}_{,y}^T(x) \mathbf{M}^{-1}(x) \mathbf{B}(x) \mathbf{u}$. The test functions derivatives satisfy the consistency conditions and are integrable so that Gauss' theorem holds. If these two requirements are fulfilled the method passes the patch test [76].

5.3 Element Free Galerkin (EFG)

The EFG uses MLS interpolants to construct the trial and test functions [18]. In contrast to the DEM certain key differences are introduced in the implementation to increase the accuracy. These differences to the DEM are [18]:

- Certain terms in the derivatives of the interpolants—which were omitted in the DEM—are included, i.e. the derivatives are computed according to Eq. 4.2.
- A much larger number of integration points has been used, arranged in a cell structure.
- EBCs are enforced “correctly”; in the first publication in [18] by Lagrangian multipliers.

The partial derivatives of the shape functions $\Phi(\mathbf{x})$ are obtained by applying the product rule to $\Phi = \mathbf{p}^T(x) \mathbf{M}^{-1}(x) \mathbf{B}(x)$ which results into

$$\Phi_{,j} = [\mathbf{p}_{,j}^T(x) \mathbf{M}^{-1}(x) \mathbf{B}(x) + \mathbf{p}^T(x) \mathbf{M}_{,j}^{-1}(x) \mathbf{B}(x) + \mathbf{p}^T(x) \mathbf{M}^{-1}(x) \mathbf{B}_{,j}(x)],$$

with $\mathbf{M}_{,i}^{-1} = -\mathbf{M}^{-1}\mathbf{M}_{,i}\mathbf{M}^{-1}$ [18]. In the DEM only the first expression in the sum has been considered but for accurate results, the coefficients $\mathbf{a}(\mathbf{x})$ should not be assumed to be constant, thus $\mathbf{p}^T(x)\mathbf{M}_{,j}^{-1}(x)\mathbf{B}(x) + \mathbf{p}^T(x)\mathbf{M}^{-1}(x)\mathbf{B}_{,j}(x)$ cannot be neglected [18].

The integration in the EFG is realized by a large number of integration points which are arranged in a cell structure, see subsection 6.2. The cells serve two purposes [18]:

- they help to identify nodes which contribute to the discrete L_2 norm at a quadrature point
- they provide a structure for the evaluation of the integrals with Gauss quadrature. The number of quadrature points depends on the number of nodes in a cell. In [18] Belytschko *et al.* have used $n_Q \times n_Q$ Gauss quadrature where $n_Q = \sqrt{m} + 2$ and m is the number of nodes in a cell.

A slightly different approach for the EFG method avoids the inversion of a matrix at every integration point. Here, weighted orthogonal basis functions are constructed for the MLS interpolants by using a Gram-Schmidt orthogonalization [103], the relation between matrix inversion and orthogonalization in the MLS is also pointed out in [17]. With the use of a weighted orthogonal basis functions, the burden of inverting a matrix \mathbf{M} at quadrature points is eliminated because the matrix becomes a diagonal matrix which is trivially invertible [103]. Mathematically we can describe this approach as follows: The matrix $\mathbf{M}(\mathbf{x})$ is diagonalized for arbitrary \mathbf{x} by imposing the following orthogonality condition at any point \mathbf{x} where $\mathbf{a}(\mathbf{x})$ is to be computed:

$$\sum_i^N w(\mathbf{x} - \mathbf{x}_i) q_k(\mathbf{x}_i) q_j(\mathbf{x}_i) = 0, \quad k \neq j.$$

For the given arbitrary basis functions $p_k(\mathbf{x})$ the orthogonal basis functions $q_k(\mathbf{x})$ can be obtained by using the Schmidt orthogonalization procedure. Because of the orthogonality condition the matrix \mathbf{M} becomes diagonal and the coefficients $\mathbf{a}(\mathbf{x})$ can be directly obtained. The advantage of using orthogonal basis functions is that it reduces the computational cost and improves the accuracy of interpolants when the matrix \mathbf{M} becomes ill-conditioned [103]. The computational costs of the orthogonalization procedure, however, are of the same order as the costs of matrix inversion. But from the viewpoint of accuracy, orthogonalization of the basis functions may be preferred over matrix inversion, since the orthogonalization procedure is equivalent with solving the

linear $k \times k$ system by means of a singular value decomposition of the moment matrix \mathbf{M} [63].

In [80] the authors claim that the EFG (=MLS) shape functions are 50-times more expensive to compute than FEM shape functions. However, if the cost for mesh generation in FEM is considered, the FEM is still faster but may be not by orders of magnitude.

5.4 Least-squares Meshfree Method (LSMM)

Every partial differential equation under consideration might be used in a weak form such that the least-squares error of the problem is minimized

$$\int_{\Omega} (\mathcal{L}u - f)^2 d\Omega \longrightarrow \min.$$

One may show that this is equivalent to using specific test functions in a Petrov-Galerkin setting, i.e. a setting where test and shape functions are chosen differently. These functions may be constructed in a mesh-based way, for example by the standard FEM functions, or in a meshfree way leading to LSMMs. LSMMs have been described by Park *et al.* in [117] and Zhang *et al.* in [131].

The least-squares formulation of a problem has a number of well-known distinct properties compared to Bubnov-Galerkin settings, see e.g. [70]. One of the advantages of numerical methods approximating the least-squares weak form is that stabilization of nonself-adjoint problems (e.g. "convection problems") is not required. A disadvantage is the higher continuity requirement on the test and shape functions, which limits the usage of many FEM shape functions that are often only C^0 continuous. Note that this is not a problem with MMs as they may easily be constructed to have any desired order of continuity. We do not further describe advantages and disadvantages of the least-squares formulation and refer the interested reader to [70].

It is noteworthy that LSMMs show the property that they are highly robust with respect to integration [117], i.e. even very coarse integration may be used reliably for the evaluation of the weak form.

5.5 Meshfree Local Petrov-Galerkin (MLPG)

Global vs. Local Weak Forms Before discussing the MLPG method the concept of a local weak form shall be introduced. It has already been pointed out that a weak form is needed for the method of weighted residuals. We

separate global and local weak forms following Atluri and Shen [4]. Global weak forms involve integrals over the global domain and boundary, while local weak forms are built over local subdomains Ω_s with local boundaries.

This can easily be seen from the following example [4, 6], where we consider Poisson's equation $\nabla^2 u(\mathbf{x}) = p(\mathbf{x})$ in a global and a local weak form. Essential boundary conditions are $u = \bar{u}$ on Γ_u , imposed with the penalty method and natural BCs are $\frac{\partial u}{\partial n} = \bar{q}$ on Γ_q . This gives

$$\int_{\Omega} \Psi (\nabla^2 u^h - p) d\Omega - \alpha \int_{\Gamma_u} \Psi (u^h - \bar{u}) d\Gamma_u = 0,$$

where Ω is the global domain. After applying the divergence theorem the global symmetric weak form follows as

$$\int_{\Gamma} \Psi \bar{q} d\Gamma - \int_{\Omega} (\Psi_{,i} u_{,i}^h + \Psi p) d\Omega - \alpha \int_{\Gamma_u} \Psi (u^h - \bar{u}) d\Gamma_u = 0.$$

The same is in a local unsymmetric weak form over a local subdomain Ω_s

$$\int_{\Omega_s} \Psi (\nabla^2 u^h - p) d\Omega_s - \alpha \int_{\Gamma_u} \Psi (u^h - \bar{u}) d\Gamma_u = 0,$$

and analogously a local symmetric weak form can be reached by applying the divergence theorem to this equation:

$$\begin{aligned} \int_{\Gamma_s} \Psi u_{,i}^h n_i d\Gamma - \int_{\Omega_s} (\Psi_{,i} u_{,i}^h + \Psi p) d\Omega - \alpha \int_{\Gamma_u} \Psi (u^h - \bar{u}) d\Gamma_u &= 0 \\ \int_{\Gamma_s^*} \Psi u_{,i}^h n_i d\Gamma + \int_{\Gamma_{su}} \Psi u_{,i}^h n_i d\Gamma + \int_{\Gamma_{sq}} \Psi \bar{q} d\Gamma - \int_{\Omega_s} (\Psi_{,i} u_{,i}^h + \Psi p) d\Omega \\ - \alpha \int_{\Gamma_u} \Psi (u^h - \bar{u}) d\Gamma_u &= 0. \end{aligned}$$

Herein, Γ_s is the boundary of the local subdomain Ω_s , and Γ_s^* stands for the part of Γ_s which is in the interior of the global domain. Γ_{su} and Γ_{sq} are those parts of Γ_s lying on the boundary of the global domain where essential and natural BCs are applied respectively. Clearly, $\Gamma_s = \Gamma_s^* \cup \Gamma_{su} \cup \Gamma_{sq}$. This equation holds irrespective of the size and shape of Γ_s and the problem becomes one as if we are dealing with a localized boundary value problem over an n -dimensional sphere Ω_s [4, 6]. It is natural to choose the supports of the weighting functions $\tilde{\Omega}_i$ as the local subdomains Ω_s which is assumed in the following.

MLPG method The MLPG is a concept rather than a method itself. It can use any meshfree approximations and any convenient test function for the solution process. Atluri and Shen examine in [4] six different realizations of the MLPG concept which they restrict to “intrinsic-basis-only” approximations due to their finding that an extrinsic basis introduces problems, because their derivatives show larger oscillations or indentations than in the other meshfree interpolations [3]. These characteristics are directly related to difficulties in the numerical integration (see subsection 6.2) for the global stiffness matrix.

The MLPG works with a local weak form instead of a global weak form; the weak form is formulated over all local subdomains $\tilde{\Omega}_i$. For this type of formulation one may claim [2] that it is more natural to perform integration over these—in general regular-shaped—subdomains and their boundaries instead of using a background mesh or cell structure for integration (see subsection 6.2). This is why Atluri *et al.* claim (e.g. [4]) that the MLPG is a “truly meshfree” method whereas methods involving the global weak form often use background meshes or cell structure for integration and may only be considered “pseudo-meshfree”. It is our opinion that the aspect of local and global weak forms should not be overemphasized as it is absolutely correct to use integration over supports in a global weak form rather than introducing local weak forms as a necessary formulation for this kind of integration.

With the MLPG concept one may derive all MMs as special cases if the trial and test functions and the integration methods are selected appropriately [3]. Some specific realizations of the MLPG concept are shortly introduced in the following.

- MLPG 1: Here, the test function over $\tilde{\Omega}_i$ is the same as the weighting function w in the MLS or RKPM method for the construction of the PU. Hence it is bell shaped and zero on the local boundary, as long as $\tilde{\Omega}_i$ does not intersect with the global boundary of Ω .
- MLPG 2: The test function becomes the Dirac Delta function and thereby a collocation method results. The integrals of the local weak form vanish and the strong form of the PDE is discretized.
- MLPG 3: The test function is the same than the residual in the differential equation, using discrete least-squares. In this case analogous methods to the LSMs (subsection 5.4) result.
- MLPG 4: The test function is the modified fundamental solution to the differential equation. This MLPG is synonymous with the Local Boundary

Integral Equation (LBIE) which has been worked out for reasons of clarity in the next subsection.

- MLPG 5: The test function is the characteristic function $\chi_{\tilde{\Omega}_i}$ and thus constant over each local subdomain $\tilde{\Omega}_i$.
- MLPG 6: The test function is identical to the trial function and thus the special case of a Bubnov-Galerkin method results. The resulting method is similar to EFG and DEM but the latter work with the global weak form instead of the local. If spheres are used for the subdomains the method has also been referred to as the Method of Finite Spheres (MFS) [37].

For a short summary of the MLPG and LBIE concept, the reader is referred to [7].

5.6 Local Boundary Integral Equation (LBIE)

The LBIE is the meshfree (and local) equivalent of the conventional boundary element method (BEM). We shall therefore recall shortly some important features of the BEM.

The BEM reduces the dimensionality of the problem by one through involving the trial functions and their derivatives only in the integral over the global boundary of the domain. The BEM is based on the boundary integral equation (BIE), which can be obtained from the weak form by choosing the test functions equal to the infinite space fundamental solution of the —at least highest order— differential operator of the PDE. This restricts the usage of the BEM to the cases where the infinite space fundamental solution is available. On the global boundary either the value $u(\mathbf{x})$ or $\frac{\partial u(\mathbf{x})}{\partial n}$ is known. If some point \mathbf{y} lies on the boundary, the BIE can be used as an integral equation for the computation of the unprescribed boundary quantity, respectively. In the BEM one has to deal with strong singularities (r^{-1}) and weak singularities ($\ln r$) involved in the integrals. Therefore some integrals have to be considered in the Cauchy Principle Value (CPV) sense when the source point \mathbf{y} is located on the boundary over which the integration is carried out [122]. After solving the system of equations, defined by a full and unsymmetric matrix, the values $u(\mathbf{x})$ and $\frac{\partial u(\mathbf{x})}{\partial n}$ on the global boundary are known. The evaluation of the unknown function and its derivatives for certain single points within the domain involves the calculation of integrals over the entire global boundary, which may be tedious and inefficient [133].

Summarizing this, one can say the BEM drops the dimensionality of the problem by one, is restricted to cases where the fundamental solution is known, involves singularities in integral expressions and leads to a full and unsymmetric but rather small matrix. Due to the fact that an exact solution (the infinite space fundamental solution) is used as a test function to enforce the weak formulation, a better accuracy may be achieved in numerical calculations [133].

The objective of the LBIE method is to extend the BEM idea to meshfree applications based on a Local Boundary Integral Equation (LBIE) approach. In the LBIE, a problem with the artificial local subdomain boundaries Γ_s occurs, due to the fact that for the local equations of the boundary terms neither $u(\mathbf{x})$ nor $\frac{\partial u(\mathbf{x})}{\partial n}$ are known (as long as they are not on the global boundary). Therefore the concept of a 'companion solution' is introduced [133]. The test function is chosen to be $v = u^* - u'$, where u^* is the infinite space fundamental solution and u' is the companion solution which satisfies a certain Dirichlet problem over the subdomain Ω_s . Thereby one can cancel out the $\frac{\partial u(\mathbf{x})}{\partial n}$ in the integral over Γ_s [133]. Thus, by using the 'companion fundamental solution' or 'modified fundamental solution', no derivatives of the shape functions $\frac{\partial u(\mathbf{x})}{\partial n}$ are needed to construct the stiffness matrix for the interior nodes, as well as for those nodes with no parts of their local boundaries coinciding with the global boundary of the domain of the problem where EBCs are applied [132].

The subdomains in the LBIE are often chosen in the following way: An d -dimensional sphere, centered at \mathbf{y} , is chosen where for simplicity reasons the size of Ω_s of each interior node is chosen to be small enough such that its corresponding local boundary $\partial\Omega_s$ will not intersect with the global boundary Γ of the problem domain Ω [132]. Only the local boundary integral associated with a boundary node contains parts of the global boundary of the original problem domain [132].

The numerical integration of boundary integrals with strongly singular kernels requires special attention in the meshfree case of the LBIE where the boundary densities are only known digitally (e.g. in the case of MLS-approximation) [122]. In [122], the authors claim:

In meshfree implementations of the BIE the question of singularities has to be reconsidered, because the boundary densities are not known in a closed form any more. This is because the shape functions are evaluated only digitally at any required point. Thus, the peak-like factors in singular kernels cannot be smoothed by cancellation of divergent terms with vanishing ones in boundary densities before the numerical integration. The proposed method consists in

the use of direct limit approach and utilization of an optimal transformation of the integration variable. The smoothed integrands can be integrated with sufficient accuracy even by standard quadratures of numerical integration.

Compared to the conventional BEM, shortly described above, the LBIE method has the following advantages [5]: The stiffness matrix is sparse, the unknown variable and its derivatives at any point inside the domain can be easily calculated from the approximated trial solution by integration only over the nodes within the domain of definition of the MLS approximation for the trial function at this point; whereas this involves an integration through all of the boundary points at the global boundary in the BEM.

Compared with MMs in general the LBIE is found to have the following advantages [5]: An exact solution (the infinite space fundamental solution) is used as a test function which may give better results, no derivatives of shape functions are needed in constructing the stiffness matrix for the internal nodes as well as for those boundary nodes with no EBC-prescribed sections on their local integral boundaries (this is attractive as the calculations of derivatives of shape functions from the MLS approximation may be quite costly [132]).

5.7 Partition of Unity Methods (PUM, PUFEM, GFEM, XFEM)

Throughout this paper the Partition of Unity FEM (PUFEM) [106], Generalized FEM (GFEM) [123, 124], Extended FEM (XFEM) [19] and the Partition of Unity Methods (PUM) [11] are considered to be essentially identical methods, following e.g. [8, 9]. Thus, we do not even claim that those methods which have the term “finite element” in their name necessarily rely on a mesh-based PU (although this might have been the case in the first publications of the method). Let us consider this “element” aspect in the sense of the Diffuse Element Method (DEM, see subsection 5.2), where it has already been shown that the same shape functions that arise in the meshfree MLS context may as well be interpreted as diffuse elements. The treatment of this aspect is not consistent throughout the publications and it may as well be found that e.g. the GFEM is considered a hybrid of the FEM and PUM [123]; in contrast, other authors [8, 9]—including the authors of this paper—may consider the GFEM and PUM equivalent.

The PUMs employ an extrinsic basis $\mathbf{p}(\mathbf{x})$ in the form

$$\begin{aligned} u^h(\mathbf{x}) &= \sum_{i=1}^N \Phi_i(\mathbf{x}) \mathbf{p}^T(\mathbf{x}) \mathbf{v}_i \\ &= \sum_{i=1}^N \Phi_i(\mathbf{x}) \sum_{j=1}^l p_j^T(\mathbf{x}) v_{ij}. \end{aligned}$$

Instead of having only u_i as unknowns we have $\mathbf{p}^T(\mathbf{x}) \mathbf{v}_i$, thus l times more unknowns $\mathbf{v}_i = v_{ij} = (a_{i1}, a_{i2}, \dots, a_{il})$ and an extrinsic basis \mathbf{p} with l components. \mathbf{p} may consist of monomials, Taylor polynomials, Lagrange polynomials or any other convenient functions. For example, Babuška and Melnik in [10] use for the Helmholtz equation in one dimension the extrinsic basis $\mathbf{p}^T = [1, x, \dots, x^{l-2}, \sinh nx, \cosh nx]$ and for the vector of unknowns $\mathbf{v}_i^T = (a_{i1}, a_{i2}, \dots, a_{il})$. The set of functions $\{\Phi_i(\mathbf{x})\}$ may either be a mesh-based or meshfree PU of n -th order consistency, thus also a simple collection of FE functions may serve as a PU.

Some main features of the PUMs are:

- It can be used for enriching a lower order consistent PU to a higher order approximation [11]. Then, the shape functions of a PUM are products of PU functions and higher order local approximation functions (usually local polynomials). Thus, the consistency of the approximation can be raised. It should be noted that more unknowns per node for a certain consistency are needed in the PUM than in MLS and RKPM. For example, while in the MLS and RKPM we always have one unknown per node for any order of consistency, this is not the case in the PUM. For example $u^h(x) = \sum_{i=1}^N \Phi_i^0(x) (1, x) (a_{i1}, a_{i2})^T$ is needed to attain first order consistency in one dimension, where $\{\Phi_i^0(x)\}$ is a 0th order consistent PU, e.g. a Shepard-PU. Then, there are two unknowns a_{i1} and a_{i2} per node. Thus, higher order consistent PUs can be constructed without the need to invert higher order $k \times k$ systems of equations at every point (k refers to the size of the *intrinsic* basis) as in the MLS and RKPM, but l unknowns per node have to be introduced (l refers to the size of the *extrinsic* basis), leading to a much larger final system of equations.
- Another application of the PUM are cases, where *a priori* knowledge about the solution is known and thus the trial and test spaces can be designed with respect to the problem under consideration [8, 11, 106, 123, 124, ...]. Therefore it shall be recalled that both FEM and MMs rely on the *local*

approximation properties of polynomials, being used in the intrinsic basis. However, if we know—from analytical considerations—that the solution has locally a non-polynomial character (e.g. it is oscillatory, singular etc.), the approximation should better be done by (“handbook”) functions that are more suited than polynomials, e.g. harmonic functions, singular functions etc., to gain optimal convergence properties. The PUM enables us to construct FE spaces which perform very well in cases where the classical FEM fails or is prohibitively expensive.

For example in case of solving the Helmholtz equation with a highly oscillatory solution, the PUM can be significantly cheaper than FEM for the same accuracy, if certain trigonometric functions are included in the basis [11].

- The PUM enables one to easily construct ansatz spaces of any desired regularity, while in the FEM it is a severe constraint to be conforming. The approximation properties of the FEM are based on the local approximability and the fact that polynomial spaces are big enough to absorb extra constraints of being conforming without loosing the approximation properties. Instead, in the PUM, the smoothness of the PU enforces easily the conformity of the global space V and allows us to concentrate on finding good local approximations for a given problem [106].
- The EBCs can be implemented by choosing the local approximation spaces such that the functions satisfy the EBCs [8, 9]. In contrast, standard MMs based on the MLS or RKPM procedure without an additional extrinsic basis require special attention for the imposition of EBCs, see subsection 6.1.

The basic idea of the PUM can shortly be described as done in [11]:

Let overlapping patches $\{\Omega_i\}$ be given which comprise a cover of the domain Ω , and let $\{\varphi_i\}$ be a partition of unity subordinate to the cover. On each patch, let function spaces V_i reflect the local approximability. Then the global ansatz space V is given by $V = \sum_i \varphi_i V_i$. Local approximation in the spaces V_i can be either achieved by the smallness of the patches (*‘h-version’*) or by good properties of V_i (*‘p-version’*).

The linear dependency of the resulting equations is an important issue in PUMs (e.g. [123]), especially when the PUM spaces are based on *polynomial* local approximation spaces. The PU and the approximation space cannot be chosen

isolated from each other. There are combinations, where the local spaces multiplied by the appropriate PU functions are linearly dependent or will at least lead to an ill-conditioned matrix. For the example when a simple mesh-based PU of first order consistency is used (simple hat function PU) and the local approximation space is polynomial this will lead to linear dependency which can easily be shown. PUs of MMs are not constructed with polynomials directly but rather rational functions. However, a problem of “nearly” linear dependency remains because for the construction of meshfree PUs an intrinsic polynomial bases is used (which is the reason for the good approximation property of MMs in case of polynomial-like solutions).

A broad theoretical background for the PUMs has been developed in [8] and [9], where results for the conventional FEM may be obtained as specific subcases of the PUM.

5.8 *hp*-clouds

The *hp*-cloud method was developed by Duarte and Oden, see e.g. [43]. The advantage of this method is that it considers from the beginning the *h* and *p* enrichment of the approximation space [49]. In contrast to MLS and RKPM, the order of consistency can be changed without introducing discontinuities, hence the *p*-version of the *hp*-cloud method is smooth. The features of the PUM—mainly its enrichment ability and the ability to include *a priori* knowledge of the solution by introducing more than one unknown at a node and usage of a suitable extrinsic basis—are also valid for the *hp*-cloud method.

The approximation in the *hp*-cloud method is:

$$\begin{aligned} u^h(\mathbf{x}) &= \sum_{i=1}^N \Phi_i(\mathbf{x}) (u_i + \mathbf{p}^T(\mathbf{x}) \mathbf{v}_i) \\ &= \sum_{i=1}^N \Phi_i(\mathbf{x}) \left(u_i + \sum_{j=1}^l p_j(\mathbf{x}) v_{ij} \right). \end{aligned}$$

We cite from [49] to reflect the concept of the *hp*-cloud method:

The essential feature of the *hp*-cloud method lies in the way the approximation functions are built in order to trivially implement the *p* enrichment. In order to accomplish this task, one has to define an open covering of the domain and an associated PU. Let Ω be

the domain and Q_N an arbitrary point set. To each node x_α we associate an open set ω_α , ($\Omega \subset \bigcup_{\alpha=1}^N \omega_\alpha$), the sets ω_α being the clouds. A set of functions $\wp_N = \{\psi_\alpha\}_{\alpha=1}^N$ is called PU subordinated to the open covering $\mathfrak{I}_N = \{\omega_\alpha\}_{\alpha=1}^N$ of Ω if the following holds: $\sum_{\alpha=1}^N \psi_\alpha(x) = 1$ for all $x \in \Omega$, $\psi_\alpha \in C_0^s(\omega_\alpha)$ with $s \geq 0$ and $\alpha = 1, 2, \dots, N$. The latter property implies that the functions ψ_α are non-zero only over the clouds ω_α . In case of the FE PU, a node x_α is a nodal point of the FE mesh and a cloud ω_α is the union of all FEs connected to that node. A set of cloud shape functions is defined as $\Phi_i^\alpha = \psi_\alpha L_i$. A linear combination of these cloud shape functions can approximate a function u as $u_{hp} = \sum_\alpha \sum_i u_i \Phi_i^\alpha$. The functions L_i can be chosen with great freedom. The most straightforward choice are polynomial functions since they can approximate smooth functions well. However, there are better choices, e.g. harmonic polynomials for the solution of Poisson's equation.

The h -refinement of an hp -cloud discretization may consist in adding more clouds of smaller size to the covering of the domain while keeping the degree of the cloud-shape functions fixed. In the case of p -enrichment the number of clouds may be kept fixed while the polynomial degree of the functions used in the construction of the cloud shape functions is increased [49]. It should be noted that a p -refinement is very easy but leads to an increase in the condition number of the resulting global matrix [49]. For a detailed description of an hp -adaptive strategy see e.g. [41].

Whereas the original hp -cloud method is a meshfree method, in [113] a hybrid method is introduced which combines features of the meshfree hp -cloud methods with features of conventional finite elements. Here, the PU is furnished by (mesh-based) conventional lower order FE shape functions [113]. The hp -cloud idea is used to produce a hierarchical FEM where all the unknown degrees of freedom are concentrated at corner nodes of the elements. This ensures in general a more compact band structure than that arising from the conventional hierarchic form [113]. Thus, the enrichment of the finite element spaces is one on a nodal basis and the polynomial order associated with a node does not depend on the polynomial order associated with neighbouring nodes [113]. The p -convergence properties in this method differ from traditional p -version elements, but exponential convergence is attained. Applications to problems with singularities are easily handled using cloud schemes [113].

5.9 Natural Element Method (NEM)

Natural neighbour interpolation was introduced by Sibson for data fitting and smoothing [120]. It is based on Voronoi cells T_i which are defined as

$$T_i = \{x \in \mathfrak{R} : d(\mathbf{x}, \mathbf{x}_i) < d(\mathbf{x}, \mathbf{x}_j) \forall j \neq i\},$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance (Euclidean norm) between \mathbf{x}_i and \mathbf{x}_j . The so-called Sibson functions or natural neighbour functions are defined by the ratio of polygonal areas of the Voronoi diagram, hence

$$\Phi_i(\mathbf{x}) = \frac{A_i(\mathbf{x})}{A(\mathbf{x})},$$

where $A(\mathbf{x}) = T_{\mathbf{x}}$ is the total area of the Voronoi cell of \mathbf{x} and $A_i = T_i \cap T_{\mathbf{x}}$ is the area of overlap of the Voronoi cell of node i , T_i , and $T_{\mathbf{x}}$. This may also be seen from Fig. 9. The support of Sibson functions turns out to be complex: It is the intersection of the convex hull with the union of all Delaunay circumcircles that pass through node i . The shape functions are C^∞ everywhere except at the nodes where they are only C^0 . It is possible to obtain C^1 continuity there with more elaborate ideas. These Sibson functions have been used as test and shape functions in the Natural Element Method (NEM) [23, 125].

There exist also the possibility to use non-Sibsonian shape functions which was introduced by Belikov *et al.* in [13]; they take the form

$$\Phi_i(\mathbf{x}) = \frac{s_i(\mathbf{x})/h_i(\mathbf{x})}{\sum_{j=1}^M s_j(\mathbf{x})/h_j(\mathbf{x})},$$

where M is the number of natural neighbours and s_i and h_i are pictured in Fig. 9; we do not give the mathematical definitions. These interpolant has been used in a Galerkin setting in [126]. There, it is also noted that the non-Sibsonian interpolant —having very similar properties than the Sibson interpolant— may be constructed with considerably less computing time. Both interpolants share important properties such as they build PUs with linear consistency, furthermore, they are strictly positive and have Kronecker delta property [126]. The latter ensures that EBCs may be imposed easily, however, it is noted in [125, 126] that non-convex domains require special attention.

One may summarize the NEM as a method that employs Sibson and non-Sibson natural neighbour-based interpolates in a Galerkin setting. A mesh is not required for the construction of the interpolants. A coupling of the NEM with the FEM is discussed in [126].

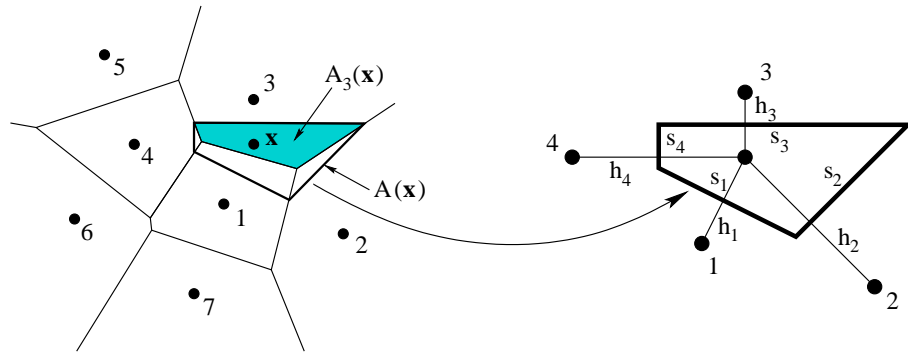


Figure 9: Construction of the Sibson and non-Sibsonian interpolant. The nodes 1, 2, 3 and 4 are called natural neighbours of \mathbf{x} .

5.10 Meshless Finite Element Method (MFEM)

The Meshless Finite Element Method was proposed in [68, 69]. The method is motivated as follows: In MMs the connectivity between nodes can always be discovered bounded in time. However, the time for the generation of a mesh as a starting point of a mesh-based simulation may *not* be bounded in time. That is, although automatic mesh generators may find some mesh, it is not guaranteed that the mesh quality is sufficient for convergence. Especially in 3D automatic mesh generation, undesirable large aspect ratio elements with almost no area/volume (“slivers”) may result, degrading the convergence rate considerably. The procedure of identifying and repairing these elements —often involving manual overhead— may require an unbounded number of iterations.

The aim of the MFEM is to obtain a good mesh connectivity in reasonable time. This cannot be reached with standard meshing procedures such as the Standard Delaunay Tessellation, which may encounter singularities for certain node situations or lead to a non-unique partition of the domain. In the MFEM context the domain is uniquely divided into polyhedra (the FEs) due to the ‘Extended Delaunay Tessellation’. These polyhedra may be arbitrarily, i.e. they are not restricted to be triangulars, quadrilaterals etc. One involves meshfree ideas to compute the shape functions on the arbitrary elements.

In the MFEM, based on Voronoi diagrams, shape functions inside each polyhedron are determined using non-Sibsonian interpolation [69], see subsection 5.9. The shape functions share Kronecker delta property. They are rather simple and

reduce for certain cases to the standard linear FEM shape functions. Consequently, only low-order quadrature rules are necessary in the MFEM leading to a very efficient method.

One may argue whether or not this method is meshfree or not. The originators of the MFEM claim in [68] that this method can as well be seen as a finite element method using elements with different geometric shapes. Meshfree ideas are only considered in the sense of finding shape functions of the arbitrary elements.

5.11 Reproducing Kernel Element Method (RKEM)

The RKEM was recently introduced by Li, Liu, Han *et al.* in a series of four papers [96, 88, 102, 121] as a hybrid of the traditional finite element approximation and the reproducing kernel approximation technique. It may be considered as an answer to the question how to find arbitrarily smooth finite element interpolations. This old problem is addressed and discussed mainly in [88], we may summarize from there that even C^1 continuous elements —needed for the simulation of 4th order boundary value problems— are difficult to obtain in the standard FEM.

The smoothness of the RKEM interpolation is achieved by involving RKPM ideas as outlined in subsection 4.3. Kronecker delta property is maintained in the RKEM, thereby simplifying the imposition of EBCs —which requires special attention for MMs, see subsection 6.1. The construction of the RKEM interpolation may be summarized as follows:

Firstly, the concept of global partition polynomials is introduced. These are mainly standard finite element functions *that are extrapolated throughout the whole domain*; we write $N_{e,i}(\mathbf{x})$ for the standard FE functions and $N_{e,i}^*(\mathbf{x})$ for the globalized functions with $e \in \Lambda_{el} = \{1, 2, \dots, n_{el}\}$, $i \in \Lambda_{ne} = \{1, 2, \dots, n_{ne}\}$ and n_{el} being the number of elements and n_{en} the number of nodes per element. The global polynomials $N_{e,i}^*(\mathbf{x})$ are C^∞ continuous in Ω . One may take the viewpoint that the multiplication of the global partition polynomials with the characteristic (Heaviside) function of an element e , which is

$$\chi_e(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega_e \\ 0, & \mathbf{x} \notin \Omega_e \end{cases},$$

leads in general to the standard truncated FE shape functions of that element, i.e. $N_{e,i}(\mathbf{x}) = N_{e,i}^*(\mathbf{x}) \chi_e(\mathbf{x})$. These local functions $N_{e,i}(\mathbf{x})$ are only C^0 continuous, which is the same order of continuity of the Heaviside function. For a

standard FEM approximation one may write

$$\begin{aligned} u(\mathbf{x}) &= \sum_{j=1}^n N_j(\mathbf{x}) u(\mathbf{x}_j) = \sum_{e \in \Lambda_{el}} \sum_{i \in \Lambda_{ne}} N_{e,i}(\mathbf{x}) u(\mathbf{x}_{e,i}) \\ &= \sum_{e \in \Lambda_{el}} \sum_{i \in \Lambda_{ne}} N_{e,i}^*(\mathbf{x}) \chi_e(\mathbf{x}) u(\mathbf{x}_{e,i}). \end{aligned}$$

The idea of the RKEM is to replace the Heaviside function by a smooth kernel function in order to obtain higher continuity of the resulting interpolation

$$u(\mathbf{x}) = \sum_{e \in \Lambda_{el}} \left[\int_{\Omega_e} K(\mathbf{x} - \mathbf{y}) d\Omega \left(\sum_{i \in \Lambda_{ne}} N_{e,i}^*(\mathbf{x}) u(\mathbf{x}_{e,i}) \right) \right].$$

The kernel is evaluated in a way that consistency of the interpolation is maintained. The same methodology as shown for the RKPM is used for this purpose, including the idea of a correction function and the solution of a small system of equations in order to obtain consistency. The continuity of the resulting interpolation only depends on the continuity of the involved window functions which localizes the global partition polynomials.

The resulting shape functions of the RKEM are considerably more complex than standard FEM shape functions, see [96, 88, 102, 121] for graphical representations of the smooth but pretty oscillatory functions. This clearly leads a large number of integration points in order to evaluate the weak form of a problem. In [121], numerical experiments in two dimensions have been performed with up to 576 quadrature points per element.

Finally, it should be mentioned that there is a relation between the Moving Particle Finite Element Method (MPFEM), introduced by Hao *et al.* [61, 62] and the RKEM. The concept of globalizing element shape functions and employing RKPM ideas to obtain consistency is also part of the MPFEM. However, it was mentioned in [96] that the nodal integration instead of full integration (see subsection 6.2) leads to numerical problems in the MPFEM.

5.12 Others

The number of MMs reviewed in this paper must be limited in order to keep it at reasonable length. We considered most methods which are mentioned and listed again and again in the majority of the publications on MMs. So it is

our belief that we hopefully covered what people mean when they use the term “meshfree methods”.

We exclude all methods that have been constructed for certain specific problems —e.g. for fluid problems— like the Finite Volume Particle Method (FVPM), the Finite Mass Method (FMM), moving particle semi-implicit method (MPS) etc. Also meshfree methods from the area of molecular dynamics (MD), the Generalized Finite Difference Method (GFDM), Radial Basis Functions (RBF), Local Regression Estimators (LRE) and Particle in Cell methods (PIC) are not considered. Although all these and many other methods are meshfree in a sense, we believe that they do not directly fit into the concept of this paper although relations undoubtedly exist.

5.13 Summary and Comments

Every MM can be classified based on the construction of a PU and the choice of the trial functions (approximation) and the test functions. This can also be seen from the overview of MMs in section 3, where all of the MMs discussed in subsection 5.1-5.11 can be found. It was also our aim to mention the starting points and individual aspects of each method.

SPH with its modified versions and MLPG 2 —among others— belong to the collocation MMs, where the test function equals the Dirac delta function. These methods solve the strong form of the PDE and do not need integrals to be evaluated as in the other weighted residual methods. This makes them fast and easy to implement. Their problems are accuracy and stability. Accuracy depends also on the choice of the dilation parameter ρ and is problem dependent. We found in subsection 4.3 that the rises and falls in error plots depending on the choice of ρ are particularly strong for collocation MMs, see Fig. 4 on page 40, and it is not possible to predict which ρ gives sufficiently good results. Stability and oscillations can be a problem for certain particle distributions and a number of ideas have been developed to fix this disadvantage.

DEM, EFG, MFS and MLPG 6 belong to the Bubnov-Galerkin MMs where the test functions equal the trial functions. Here we find good accuracy and convergence being less sensitive to the choice of ρ (see subsection 4.3). This does not hold for the DEM which is the earliest version of this class of MMs including major simplifications. The problem of these methods is the computational burden associated with the numerical evaluation of the integrals in the weak form.

MLPG 5 chooses the test function to be the characteristic function in the trial support or any smaller support. This often makes shifting of the volume in-

tegrals onto surface integrals via the divergence theorem for most of the integral expressions in the weak form possible and thereby reduces the dimension of the integration domain by one. This can save computing time significantly. However, own experiences gave unsatisfactory results for many problems, including advection-diffusion, Burgers and Navier-Stokes equations. The authors of this method, Shen and Atluri in [4], obtained good results for Poisson's equation but did not use it for the solution of flow related problems.

LSMMs solve the least-squares weak form of a problem. Advantages and disadvantages of these methods are well known, see e.g. [70]. It has been found that LSMMs are considerably more robust with respect to the accuracy of the integration, less integration points are needed for suitable results.

GFEM, XFEM, PUM, PUFEM and *hp*-clouds are based on the concept of an extrinsic basis. Thereby the order of consistency of an existing PU can be raised or *a priori* knowledge about the solution can be added to the solution spaces. The final system of equations becomes significantly larger. In practice these methods proved to be successful in very special cases (like the solution of the Helmholtz equation).

NEM and MFEM rely on shape functions which are constructed based on Voronoi diagrams (Sibson and non-Sibsonian interpolations). They do not take the MLS/RKPM way to obtain a certain consistency. It seems to the authors of this paper that the use of Voronoi diagrams as an essential part of the method is already something in-between meshfree and mesh-based. This becomes obvious in the MFEM, which may either be interpreted in a mesh-based way as a method which employs general polygons as elements, or in a meshfree way because rather the Voronoi diagram is needed than an explicit mesh. So one might say that the procedure in a mesh-based method is: node distribution \rightarrow Voronoi diagram \rightarrow mesh \rightarrow shape functions. In the NEM and MFEM only the mesh step is skipped, whereas in standard MMs based on the MLS/RKPM concepts we only have the steps: node distribution \rightarrow shape functions.

Concerning the RKEM, one may expect that the complex nature of the shape functions in this method will anticipate a breakthrough of this approach in practice. At least this method provides an answer to the question of how to find continuous element interpolations. Simple approaches are not available and the complexity of the RKEM approximations might be the necessary price to pay.

There are also other MMs which rely on the choice of other certain test functions which will not be discussed further. Also, it is impossible to even mention *every* MM in this paper, however, most of the important and frequently discussed MMs should be covered.

6 Related Problems

6.1 Essential Boundary Conditions

Due to the lack of Kronecker delta property of most of the meshfree shape functions the imposition of EBCs requires certain attention. A number of techniques have been developed to perform this task. One may divide the various methods in those that modify the weak form, those that employ shape functions with Kronecker delta property along the essential boundary and others. The first class of methods is described in subsections 6.1.1 to 6.1.4, the second from 6.1.5 to 6.1.8 and other methods not falling into these two classes in 6.1.9 and 6.1.10. We do only briefly describe the methods mentioning some of their important advantages and disadvantages, the interested reader is referred to the references given below..

It is our impression that the imposition of EBCs in MMs is only a solved problem in the sense that it is easily possible to fulfill the prescribed boundary values directly at the nodes. However, as e.g. noted in [60], a degradation in the convergence order may be found for most of the imposition techniques in two or more dimensions for consistency orders higher than 1.

6.1.1 Lagrangian Multipliers

A very common approach for the imposition of EBCs in MMs is the Lagrangian multiplier method. It is well-known that in this case the minimization problem becomes a saddle problem [24]. This method is also used in many other applications of numerical methods (not related to MMs); therefore, it is not described here in further detail.

The Lagrangian multiplier method is a very general and accurate approach [17]. However, Lagrangian multipliers need to be solved in addition to the discrete field variables, and a separate set of interpolation functions for Lagrangian multipliers is required. This set has to be chosen carefully with respect to the Babuška-Brezzi stability condition [24], which influences the choice of interpolation and the number of used Lagrangian multipliers. In addition to the increase in the number of unknowns the system structure becomes awkward, i.e. it becomes $\begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^T & 0 \end{bmatrix}$ instead of only $[\mathbf{K}]$. This matrix is not positive definite and possesses zeros on its main diagonal and solvers taking advantage of positive definiteness cannot be used any longer [18, 103]. Especially for dynamic and/or nonlinear problems (e.g. [25]) this larger system has to be solved at each

time and/or incremental step (in nonlinear problems, incremental and iterative procedures are required).

6.1.2 Physical Counterpart of Lagrangian Multipliers

In many physical problems the Lagrangian multipliers can be identified with physical quantities [103]. For example in elasticity, one can show with help of Gauss's divergence theorem that the solution for the Lagrangian multipliers λ_i of the weak form can be identified with the stress vector t on Γ_u , i.e. $\lambda_i = \sigma_{ij}n_j = t_i$ on Γ_u [63]. Or in heat conduction problems the Lagrange multiplier can be identified with the boundary flux [63].

Thus, a modified variational principle can be established in which the Lagrangian multipliers are replaced at the outset by their physical counterpart [103]. The advantage of this idea is that the modified variational principle results in a positive definite, sparse matrix. The disadvantage on the other hand is the somewhat reduced accuracy and the inconvenience compared to direct imposition of EBCs [75]. The Lagrange multiplier implementation is more accurate, but the accuracy can be equaled by adding approximately 25 – 50% more nodes [103].

It should be mentioned that such a modified variational principles tend not to work very well with FEM, particularly those of low order, because the implicit Lagrange multiplier is a lower order field than the variable which is constrained [103]. However, in MMs this modified variational principle appears to perform quite well for reasonable number of unknowns.

6.1.3 Penalty Approach

EBCs can be weakly imposed by means of a penalty formulation, where a penalty term of the form

$$\alpha \int \Psi(u_i - \bar{u}_i) d\Gamma$$

with $\alpha \gg 1$ is added to the weak form of the problem, see e.g. [112]. The success of this method is directly related to the usage of large numbers for α . This on the contrary influences the condition number of the resulting system of equations in a negative way, i.e. the system is more and more ill-conditioned with increasing values for α . The advantages of the penalty approach is that the size of the system of equations is constant and the possible positive definiteness remains for large enough α .

6.1.4 Nitsche's Method

The Nitsche method may be considered a consistent improvement of the penalty method [46]. That is, rather than adding only one term to the weak form of the problem a number of terms is added depending on the specific problem under consideration. The α -value may be chosen considerably smaller than in the Penalty method —avoiding an ill-conditioning—, and the advantages of the Penalty method remain. Therefore, it is claimed in [46] and [9] that the Nitsche method is superior to both the penalty method and Lagrange multiplier method.

6.1.5 Coupling with Finite Elements

Any of the coupling methods to be discussed in subsection 6.3 may be used to employ a string of elements along the essential boundaries and to combine the FE shape functions defined on this string with the meshfree approximation, see Fig. 10. This idea was first realized in [75] based on the ramp function approach of Belytschko [20] (subsection 6.3.1). The coupling approach of Huerta [65] (subsection 6.3.2), working with modified consistency conditions and the bridging scale method [129] (subsection 6.3.3) were applied for the purpose of imposing EBCs in [66]. There, it is found that the bridging scale method is not advisable for this purpose, due to the fact that the shape functions in this method only vanish at the boundary nodes but not along the element edges in 2D or element surfaces in 3D.

The advantage of this approach is clearly that all shape functions related to the essential boundary have Kronecker delta property as they are standard FEM functions and EBCs may be easily imposed. The disadvantage is that a string of elements has to be generated, and that the coupling necessarily leads to a somewhat complicated code structure.

A closely related approach, being a mixture of the Huerta approach and the bridging scale idea (enrichment), is presented by Chen *et al.* in [26] where no elements are needed any longer. Instead, rather arbitrary functions with Kronecker delta property may be chosen for the boundary nodes and consistency is ensured by enrichment functions. The advantage is that no string of elements is needed any longer, however, the problem is that EBCs can only be applied exactly directly at the boundary node which has already been shown problematic in the bridging scale framework [66].

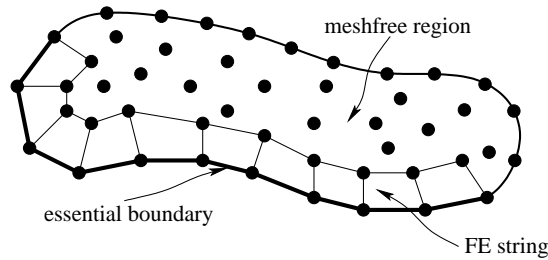


Figure 10: Usage of a finite element string along the essential boundary for imposition of EBCs.

6.1.6 Transformation Method

There exist a full and a partial (boundary) transformation method, see e.g. [25, 29, 87]. In the first an inversion of a $N \times N$ matrix is required and Kronecker Delta property is obtained at all nodes, and in the latter only a reduced system has to be inverted and Kronecker delta property is obtained at boundary nodes only. It has been mentioned in [29] that the transformation methods are usually used in conjunction with Lagrangian kernels, see subsection 4.5, because then the matrix inversion has to be performed only once at the beginning of the computation.

The basic idea of the full transformation method is as follows: The relation between the (real) unknown function values $u^h(\mathbf{x}_j)$ and the (fictitious) nodal values \hat{u}_i for which we solve the global system of equations is

$$\begin{aligned} u^h(\mathbf{x}_j) &= \sum_{i=1}^N \Phi_i(\mathbf{x}_j) \hat{u}_i, \\ \mathbf{u} &= \mathbf{D} \hat{\mathbf{u}} \end{aligned} \quad (6.1)$$

which follows directly when the approximation $u^h(\mathbf{x}) = \sum_i \Phi_i(\mathbf{x}) \hat{u}_i$ is evaluated at all nodal positions \mathbf{x}_j for $j = 1, \dots, N$. The final system of equations which results from a meshfree procedure is $\mathbf{A} \hat{\mathbf{u}} = \mathbf{b}$. However, boundary conditions are prescribed for the (real) nodal values \mathbf{u} instead of $\hat{\mathbf{u}}$. Therefore, $\hat{\mathbf{u}}$ is replaced according to Eq. 6.1 by $\mathbf{D}^{-1} \mathbf{u}$ and for the final system of equation follows

$$\mathbf{A} \mathbf{D}^{-1} \mathbf{u} = \mathbf{b},$$

and the EBCs can directly be applied. One may also interpret the shape functions $\mathbf{\Upsilon}^T = \mathbf{\Phi}^T \mathbf{D}^{-1}$ as the transformed meshfree shape functions having Kronecker delta property. It is important to note that these transformed functions are not local any longer as \mathbf{D}^{-1} is a full matrix.

The partial transformation method only requires a matrix inversion of size $N_\Gamma \times N_\Gamma$ where N_Γ is the number of nodes where EBCs are to be prescribed. The idea is as follows: One separates the particle sets of N_Ω nodes in the interior of the domain and N_Γ nodes on the boundary, clearly, $N = N_\Omega + N_\Gamma$. For the approximation follows (we omit \wedge in the following)

$$\begin{aligned} u(\mathbf{x}) &= \sum_{i=1}^N \Phi_i(\mathbf{x}) u_i \\ &= \sum_{i=1}^{N_\Omega} \Phi_i^\Omega(\mathbf{x}) u_i^\Omega + \sum_{i=1}^{N_\Gamma} \Phi_i^\Gamma(\mathbf{x}) u_i^\Gamma \\ &= \mathbf{\Phi}^\Omega \mathbf{u}^\Omega + \mathbf{\Phi}^\Gamma \mathbf{u}^\Gamma. \end{aligned}$$

This results into N_Γ equations where EBCs $u(\mathbf{x}_j) = g(\mathbf{x}_j)$ for $j = 1, \dots, N_\Gamma$ are prescribed

$$\begin{aligned} u(\mathbf{x}_j) &= \underbrace{\sum_{i=1}^{N_\Omega} \Phi_i^\Omega(\mathbf{x}_j) u_i^\Omega}_{\mathbf{D}^\Omega \mathbf{u}^\Omega} + \underbrace{\sum_{i=1}^{N_\Gamma} \Phi_i^\Gamma(\mathbf{x}_j) u_i^\Gamma}_{\mathbf{D}^\Gamma \mathbf{u}^\Gamma} = \underbrace{g(\mathbf{x}_j)}_{\mathbf{g}} \\ &= \underbrace{(N_\Gamma \times N_\Omega)(N_\Omega \times 1)}_{\mathbf{D}^\Omega \mathbf{u}^\Omega} + \underbrace{(N_\Gamma \times N_\Gamma)(N_\Gamma \times 1)}_{\mathbf{D}^\Gamma \mathbf{u}^\Gamma} = \underbrace{(N_\Gamma \times 1)}_{\mathbf{g}}, \\ &\Rightarrow \mathbf{u}^\Gamma = [\mathbf{D}^\Gamma]^{-1} (\mathbf{g} - \mathbf{D}^\Omega \mathbf{u}^\Omega). \end{aligned}$$

Inserting this into the approximation gives

$$\begin{aligned} u(\mathbf{x}) &= \mathbf{\Phi}^\Omega \mathbf{u}^\Omega + \mathbf{\Phi}^\Gamma [\mathbf{D}^\Gamma]^{-1} (\mathbf{g} - \mathbf{D}^\Omega \mathbf{u}^\Omega) \\ &= (\mathbf{\Phi}^\Omega - \mathbf{\Phi}^\Gamma [\mathbf{D}^\Gamma]^{-1} \mathbf{D}^\Omega) \mathbf{u}^\Omega + \mathbf{\Phi}^\Gamma [\mathbf{D}^\Gamma]^{-1} \mathbf{g}, \end{aligned}$$

where $\mathbf{\Upsilon}^\Omega = (\mathbf{\Phi}^\Omega - \mathbf{\Phi}^\Gamma [\mathbf{D}^\Gamma]^{-1} \mathbf{D}^\Omega)$ and $\mathbf{\Upsilon}^\Gamma = \mathbf{\Phi}^\Gamma [\mathbf{D}^\Gamma]^{-1}$ may also be interpreted as the transformed shape functions for which the EBCs can be directly applied.

6.1.7 Singular Weighting Functions

It was already realized by Lancaster and Salkauskas when introducing the MLS in [82] that singular weighting functions at all nodes recover Kronecker delta

property of the shape functions. Advantage of this has been made e.g. in [73] for the easy imposition of EBCs in a Galerkin setting.

Instead of applying singular weight functions at *all* nodes, in [29] a 'boundary singular kernel' approach is presented. Here, only the weight functions associated with constrained boundary nodes are singular. By using a singular weight function at the point \mathbf{x}_D where an EBC is prescribed, we obtain a shape function $\Phi_D(\mathbf{x}_D) = 1$ and all other shape functions at \mathbf{x}_D are $\Phi_i(\mathbf{x}_D) = 0$. Note that $\Phi_D(\mathbf{x}_i) \neq 0$, thus Φ_D is not a real interpolating function having Kronecker delta property (although $\Phi_D(\mathbf{x}_D) = 1$), because it is not necessarily 0 at all other nodes [29].

It is claimed in [63] that singular weighting functions lead to less accurate results, especially for relatively large supports. This is also due to the necessity to distribute integration points carefully such that they are not too close to the singularity which leads to ill-conditioned mass matrices. Hence, singular weighting functions are not recommended.

6.1.8 PUM Ideas

Referring to subsection 5.7, the EBCs can be implemented by choosing the local approximation spaces such that the functions satisfy the Dirichlet boundary conditions [8, 9]. For example in [17, 87] Legendre polynomials are used as an extrinsic basis recovering Kronecker delta property.

6.1.9 Boundary Collocation

This method is a simple, logical and effective strategy for the imposition of EBCs. The EBCs $u = g$ are enforced by $u(\mathbf{x}_j) = \sum_{i=1}^N \Phi_i(\mathbf{x}_j) \hat{u}_i = g(\mathbf{x}_j)$ (note again the difference between fictitious and real nodal values) [109]. This expression is taken directly as one equation in the total system of equations.

This method can enforce EBCs exactly only at boundary points but not in between these nodes [4]. However, this strategy is a straightforward generalization of the imposition of EBCs in the FEM. Compared to FEM this method reduces the higher effort of imposing EBCs in MMs in only having different elements in the matrix line which belongs to the particle (node) where EBCs are to be applied. Assume that at particle i an EBC is prescribed. Then, in the

FEM and MMs the belonging line in the matrix will look as

$$\begin{array}{ccc} \text{FEM} & & \\ \left(\begin{array}{cccc} \dots & & & \\ \dots N_{i-1}(x_i) & N_i(x_i) & N_{i+1}(x_i) & \dots \\ & \dots & & \end{array} \right) & \longrightarrow & \left(\begin{array}{ccc} \dots & & \\ \dots = 0 & = 1 & = 0 \dots \\ & \dots & \end{array} \right) \\ \text{MM} & & \\ \left(\begin{array}{cccc} \dots & & & \\ \dots \Phi_{i-1}(x_i) & \Phi_i(x_i) & \Phi_{i+1}(x_i) & \dots \\ & \dots & & \end{array} \right) & \longrightarrow & \left(\begin{array}{ccc} \dots & & \\ \dots \neq 0 & \neq 1 & \neq 0 \dots \\ & \dots & \end{array} \right). \end{array}$$

Thus, one can see the similarity of FEM and MMs in the matrix line which belongs to a node x_i where an EBCs has to be enforced. The difference can be seen in the right hand side. Here, in the FEM we know already the values of the line due to the Kronecker Delta property, whereas in MMs one has to compute all Φ at x_i . However, the idea in both methods stays the same.

It is important to note that an important condition is not fulfilled for the standard boundary collocation method: It is required that the test functions in a weak form must vanish along the essential boundary [128]. Neglecting this leads to a degradation in the convergence order, especially for meshfree shape functions with high consistency orders. Therefore, Wagner and Liu propose a corrected collocation method in [128] which considers the problem of non-vanishing test functions along the essential boundary. This idea is further considered and modified in [130].

6.1.10 D'Alembert's Principle

Using D'Alembert's principle for the imposition of EBCs was first introduced by Günther and Liu in [58]; this approach has similarities with the transformation methods (subsection 6.1.6). D'Alembert was the first to formulate a principle of replacing n differential equations of the form

$$\mathbf{f}^{inert}(\mathbf{d}, \dot{\mathbf{d}}, \ddot{\mathbf{d}}) + \mathbf{f}^{int}(\mathbf{d}, \dot{\mathbf{d}}) = \mathbf{f}^{ext} + \mathbf{f}^r$$

and m constraints

$$\mathbf{g}(\mathbf{d}) = 0$$

by $n - m$ unconstrained equations. Herein, \mathbf{f}^{inert} are inertial forces, \mathbf{f}^{int} and \mathbf{f}^{ext} are internal and external forces respectively and \mathbf{f}^r are the reaction forces which can be written as $\mathbf{f}^r = \mathbf{G}\lambda$, where $\mathbf{G}^T = \frac{\partial \mathbf{g}(\mathbf{y})}{\partial \mathbf{d}^T}$ is the constraint matrix

and λ are Lagrangian multipliers. D'Alembert asserted that, if one were to choose $n - m$ independent generalized variables \mathbf{y} such that $\mathbf{g}(\mathbf{d}(\mathbf{y})) = 0$ for all $\mathbf{y} \in \mathbb{R}^{n-m}$, then we can write instead of the formerly two equations the smaller system

$$\mathbf{J}^T (\mathbf{f}^{inert}(\mathbf{d}(\mathbf{y}), \mathbf{J}\dot{\mathbf{y}}, \mathbf{J}\ddot{\mathbf{y}}) + \mathbf{f}^{int}(\mathbf{d}(\mathbf{y}), \mathbf{J}\dot{\mathbf{y}})) = \mathbf{J}^T \mathbf{f}^{ext},$$

with $\mathbf{J} = \mathbf{J}(\mathbf{y}) = \left(\frac{\partial \mathbf{d}(\mathbf{y})}{\partial \mathbf{y}^T} \right)$ being the $n \times (n - m)$ Jacobian matrix [58]. Due to $\mathbf{J}^T \mathbf{G} = 0 \forall \mathbf{y}$ the reaction forces cancel out of the system.

With this idea it is also possible to impose BCs in Galerkin methods, hence also in MMs. Here, we might have a system of the kind [57]

$$\underbrace{\int_{\Omega} w_{\mathbf{u}}(\dots) + w_{\mathbf{u},\mathbf{x}}(\dots) d\Omega}_{\mathbf{f}^{inert} + \mathbf{f}^{int}} = \underbrace{\int_{\Gamma_h} \mathbf{w}_{\mathbf{u}} h d\Gamma}_{\mathbf{f}^{ext}} + \underbrace{\int_{\Gamma_g} \mathbf{w}_{\lambda} (u - g) d\Gamma}_{\mathbf{g}(\mathbf{d}) = 0} + \underbrace{\int_{\Gamma_g} \mathbf{w}_{\mathbf{u}} \lambda d\Gamma}_{\mathbf{f}^r}.$$

Discretization leads to a matrix equation and application of d'Alembert's principle is done analogously to the above shown case. It only remains to find a suitable $(n - m) \times 1$ vector \mathbf{y} of generalized variables. The mapping from the generalized to the nodal variables \mathbf{d} —i.e. the Jacobian matrix \mathbf{J} —can be obtained via an orthogonalization procedure, e.g. with help of the Gram-Schmidt algorithm. Consequently $\mathbf{J}^T \mathbf{G} = 0 \forall \mathbf{y}$ will be fulfilled and also $\mathbf{J}^T \mathbf{J} = \mathbf{I}$.

\mathbf{G} and \mathbf{J} can be interpreted in the shape function context as follows. We obtain $u^h = \mathbf{N}^T \mathbf{d} = \mathbf{N}^T \mathbf{J} \mathbf{y} + \mathbf{N}^T \mathbf{G} \mathbf{g}$, which can be interpreted as splitting the shape function set \mathbf{N} into the interior set $\mathbf{N}_{\mathbf{J}} = \mathbf{J}^T \mathbf{N}$ and the boundary set $\mathbf{N}_{\mathbf{G}} = \mathbf{G}^T \mathbf{N}$ [57, 58].

Summarizing this, one can state that D'Alembert's principle uses a set of generalized variables, and a Jacobian matrix to project the residual onto the admissible solution space [58].

6.2 Integration

Using the method of weighted residuals leads to the weak form of a PDE problem. The expressions consist of integration terms which have to be evaluated numerically. In MMs this is the most time-consuming part—although being parallelizable, see subsection 6.6—, as meshfree shape functions are very complicated and a large number of integration points is required in general. Not only that the functions are rational, they also have different forms in each small region $\tilde{\Omega}_{I,k}$ (see Fig. 11) where the same nodes have influence respectively. As

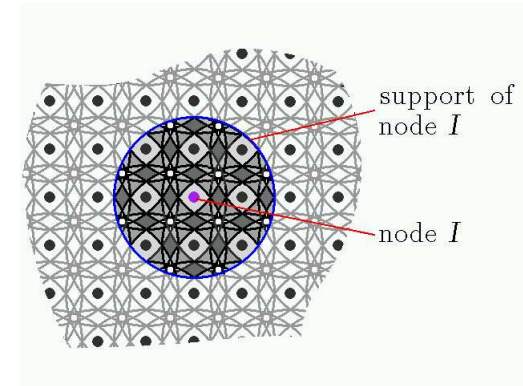


Figure 11: Overlap of circular supports in a regular two-dimensional node distribution; the support $\tilde{\Omega}_I$ of node I is highlighted. The differently colored regions $\tilde{\Omega}_{I,k}$ of this support have different influencing nodes and consequently a different rational form of the shape function.

a consequence, especially the derivatives of the shape functions might have a complex behaviour along the boundaries of each $\tilde{\Omega}_{I,k}$. In Fig. 11 each colored area stands for a region $\tilde{\Omega}_{I,k}$ of a certain support $\tilde{\Omega}_I$.

It is an important advantage of the collocation MMs that they solve the strong form of the PDE and no integral expressions have to be calculated. However, the disadvantages of these methods have been mentioned, see e.g. subsection 5.1.

Numerical integration rules Numerical integration rules are of the form

$$\int f(\mathbf{x}) d\Omega = \sum f(\mathbf{x}_i) w_i$$

and vary only with regard to the locations \mathbf{x}_i and weights w_i of the integration points (note that the integration weights w_i have nothing to do with the MLS weighting function or RKPM window function). Available forms include for example Gaussian integration and trapezoidal integration. (Monte-Carlo integration may also be considered as an interpretation of “integration” in collocation MMs).

Gaussian integration rules are most frequently used for the integration in MMs. They integrate polynomials of order $2n_q - 1$ exactly where n_q is the

number of quadrature points. The special weighting of this rule makes only sense if the meshfree integrands are sufficiently polynomial-like in the integration domains. That means if the integration domains in which the integration points are distributed according to the Gauss rule are small enough such that the rational, sometimes non-smooth character of the meshfree integrands is of less importance, then suitable results may be expected.

Otherwise, if the rational non-smooth character of the integrand is of importance, e.g. in the case where the integrand (being a product of a test and trial function) is zero in part of the integration area, the trapezoidal rule may be preferred.

Accuracy of the integration The accuracy of the numerical integration of the weak form of the problem under consideration is crucial to the resulting accuracy of the method. It is claimed in [8] that in order to maintain the optimal order of convergence, the numerical quadrature rule should approximate the elements of the final matrix with a relative accuracy of $O(h^{n+2})$. This property, however, is difficult to prove for any of the integration rules because the integrands in MMs are rational and not necessarily smooth in the integration domain.

MMs in general enjoy comparably high convergence properties compared to mesh-based FEM approximations with an equivalent basis. It may often be found in numerical experiments with MMs that the convergence order is very high for coarse to moderate particle distributions but flattens for very large numbers of particles due to the reason that the number of integration points is not sufficiently increased. Using an n -th order integration rule to obtain m -th order convergence with $m > n$ for the whole possible range of particles numbers is not possible if the number of integration domains with a constant number of integration points is directly proportional to the number of particles. Thus, the number of integration domains must be increased considerably faster than the particle number, or keeping this ratio constant the number of integration points per integration domain must be increased.

Another issue is the divergence theorem. It is sometimes claimed that MMs do not pass the patch test exactly. This is related to the following consideration: A weak of a PDE under consideration is often treated with the divergence theorem which shifts derivatives between test and trial functions, e.g.

$$\int w N_{,xx} dx = \int w_{,x} N_{,x} + \text{boundary terms.}$$

This, however, assumes an exact integration which is not possible for the rational test and trial functions of MMs. Therefore, the divergence theorem is “not fully correct” —by means of the integration error— with the consequence that the patch test is not longer exactly fulfilled (which is related to a loss of consistency). The patch test may thus only be exactly (with machine precision) fulfilled as long as the problem is given in the so-called Euler-Lagrange form

$$\int_{\Omega} w (\mathcal{L}u - f) d\Omega = 0,$$

however, not after manipulations with the divergence theorem. In the Euler-Lagrange form the integration error plays no role and the patch test is fulfilled exactly for all numerical integration rules.

In spite of these remarks it is in general not difficult in practice to employ integration rules that lead to reasonable accurate solutions. In the following, approaches for the numerical integration of the weak form in MMs are described.

6.2.1 Direct Nodal Integration

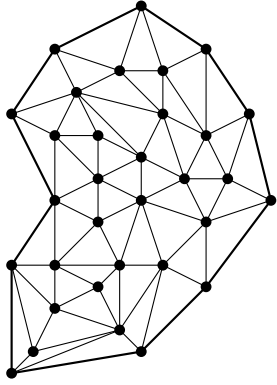
Evaluating the integrals only at the nodal positions \mathbf{x}_i instead of introducing integration points \mathbf{x}_Q is called direct nodal integration. The integration is clearly substantially more efficient than using full integration. However, in addition to comparatively large integration errors a stability problem arises for the direct nodal integration which is very similar to the numerical problems in collocation methods such as SPH, see subsection 5.1. It has been pointed out in [14, 15] that SPH collocation and Galerkin methods with nodal integration are equivalent in most cases. Therefore, the similar ideas as for the SPH may be used to stabilize the nodal integration. For example stress points and least-squares stabilization terms may be used [14]. In [30] and in [12] stabilized nodal integration techniques have been proposed, the first employing the concept of strain smoothing, the latter the concept of consistent least-squares stabilization.

Even for stabilized nodal integration schemes accuracy —in reference to convergence rate and absolute accuracy— of nodally integrated Galerkin MMs is considerably lower than for full integration, see e.g. [15] for a comparison.

6.2.2 Integration with Background Mesh or Cell Structure

Here, the domain is divided into integration domains over which Gaussian quadrature is performed in general. The resulting MMs are often called pseudo-meshfree as only the approximation is truly meshfree, whereas the integration

integration with background mesh



integration with cell structure

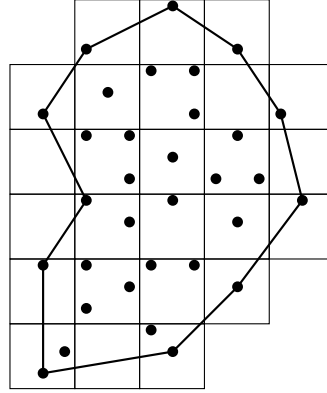


Figure 12: Integration with background mesh or cell structure.

requires some kind of mesh. In case of a background mesh, nodes and integration cell vertices coincide in general —as in conventional FEM meshes—, however, it is important to note that the background mesh does *not* have to be conforming and hanging nodes may easily be employed. In case of integration with a cell structure, nodes and integration cell vertices do in general not coincide at all [39]. This is depicted in Fig. 12.

The problem of background meshes and cells is that the integration error which arises from the misalignment of the supports and the integration domains is often higher than the one which arises from the rational character of the shape functions [39]. Accuracy and convergence are thus affected mostly from this misalignment and it might be possible that even higher order Gaussian rules do not lead to better results [39]. Note that in case of the FEM supports and integration domains always coincide.

In [39] a method has been presented to construct integration cells with a bounding box technique that align with the shape functions supports. Thereby the integration error can be considerably lowered and although more integration cells arise, the usage of lower order Gaussian quadrature is possible (3×3 Gaussian rule is suggested in [39]). Thus, more integration cells does not automatically mean more integration points. With rising order of Gauss rule the error is reduced monotonically. This approach is very closely related to integration over intersections of supports as discussed in subsection 6.2.3.

Here, it shall be recalled that the support of the shape function $\Phi_i(\mathbf{x})$ is equivalent to the support of the weighting function $w_i(\mathbf{x})$ and can be of arbitrary shape. But for the proposed bounding box method parallelepipeds —producible by tensor product weighting functions, see subsection 4.5— must be taken as support regions, because the alignment of integration cells with *spherical* supports is almost impossible [39] (see e.g. Fig. 11). Therefore tensor product based supports have to be used here, because then the overlapping supports construct several polygons, for which integration rules are readily available.

Griebel and Schweitzer go the same way in [53] using sparse grid integration rules [51] in the intersecting areas.

The use of adaptive integration by means of adaptively refining the mesh (which does not have to be conforming) or cell structure has been shown in [123].

6.2.3 Integration over Supports or Intersections of Supports

This method is a natural choice for the MLPG methods based on the local weak form but may also be used for any other of the Galerkin MMs. The resulting scheme is truly meshfree. The domain of integration is directly the support of each node or even each intersection of the supports respectively. The results in the latter case are much better than in the classical mesh or cell-based integration of the pseudo-meshfree methods for the same reason as in the above mentioned closely related alignment technique.

From an implementational point of view it should be mentioned that the resulting system matrix is integrated line by line and no element assembly is employed. For the integration over supports the integration points are distributed individually for each *line* of the final matrix, whereas the integration over intersection of supports distributes integration points for each *element* of the final matrix individually.

Special Gauss rules and mappings can be used to perform efficient integration also for spheres intersecting with the global boundary of the domain, thereby being not regular any longer, see e.g. [37]. The principle is shown in Fig. 14.

6.3 Coupling Meshfree and Mesh-based Methods

It is often desirable to limit the use of MMs to some parts of the domain where their unique advantages —meshfree, fast convergence, good accuracy, smooth

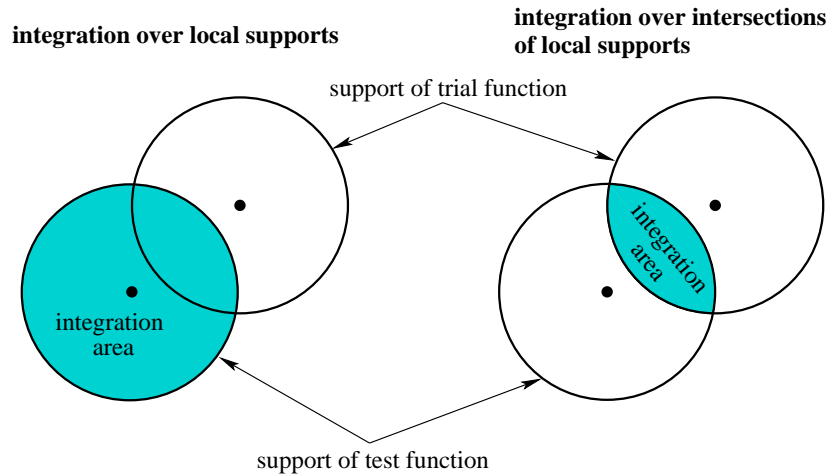


Figure 13: Integration over local supports or intersections of local supports.

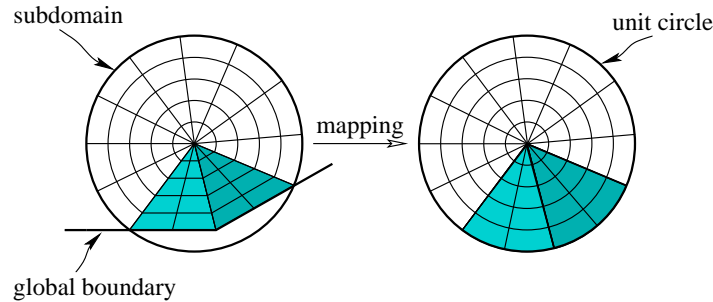


Figure 14: Mapping of special integration situations in order to apply standard integration rules.

derivatives, trivial adaptivity— are beneficial. This is because often the computational burden of MMs is much larger than in conventional mesh-based methods, thus coupling can save significant computing time. The objective is always to use the advantages of each method.

We only refer to coupling procedures where coupled shape functions result. Physically motivated *ad hoc* approaches— often aiming at conservation of mass, volume and/or momentum— such as those used for the coupling of FEM and SPH, see [72] and references therein, without accounting for consistency aspects etc. are not further considered herein.

There are several methodologies to couple meshfree and mesh-based regions.

6.3.1 Coupling with a Ramp Function

Coupling with a ramp function was introduced by Belytschko *et al.* in [20]. The domain Ω is partitioned into disjoint domains Ω^{el} and Ω^{MM} with the common boundary Γ^{MM} . Ω^{el} is discretized by standard quadrilateral finite elements and is further decomposed into the disjoint domains Ω^* —being the union of all elements along Γ^{MM} , also called transition area— and the remaining part Ω^{FEM} , connected by a boundary labeled Γ^{FEM} ; clearly $\Omega^{\text{FEM}} \cap \Omega^{\text{MM}} = \emptyset$. This situation is depicted in Fig. 15.

The mesh-based approximation $u^{\text{FEM}}(\mathbf{x}) = \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) u(\mathbf{x}_i)$ is defined in Ω^{el} , i.e. complete bilinear shape functions are defined over all elements. The meshfree approximation $u^{\text{MM}}(\mathbf{x}) = \sum_{i \in I^{\text{MM}}} \Phi_i(\mathbf{x}) u(\mathbf{x}_i)$ may be constructed for *all* nodes in Ω with meshfree shape functions as they for example arise in the MLS method. However, one may also restrict the nodes I^{MM} where meshfree shape functions are employed to at least $\Omega^{\text{MM}} \cup \Omega^*$.

The resulting coupled approximation according to the ramp function method is defined as [20]

$$\begin{aligned} u_i^h(\mathbf{x}) &= u_i^{\text{FEM}}(\mathbf{x}) + R(\mathbf{x}) [u_i^{\text{MM}}(\mathbf{x}) - u_i^{\text{FEM}}(\mathbf{x})] \\ &= (1 - R(\mathbf{x})) u_i^{\text{FEM}}(\mathbf{x}) + R(\mathbf{x}) u_i^{\text{MM}}(\mathbf{x}), \end{aligned}$$

where $R(\mathbf{x})$ is the ramp function. It is defined by using the FE bilinear shape functions as $R(\mathbf{x}) = \sum_{i \in I^*} N_i(\mathbf{x})$, $I^* = \{i : \mathbf{x}_i \in \Gamma^{\text{MM}}\}$, meaning that the ramp function is equal to the sum of the FE shape functions associated with interface element nodes that are on the boundary Γ^{MM} . Consequently, $R(\mathbf{x}) = 1$ on the boundary Γ^{MM} towards the MM region and $R(\mathbf{x}) = 0$ on the boundary Γ^{FEM} towards the FEM region and varies monotonously in between the interface region.

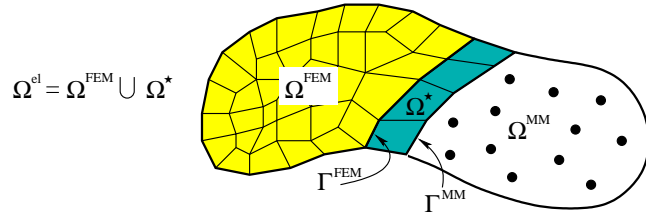


Figure 15: Coupling meshfree and mesh-based methods: Decomposition of the domain into Ω^{FEM} , Ω^{MM} and Ω^* .

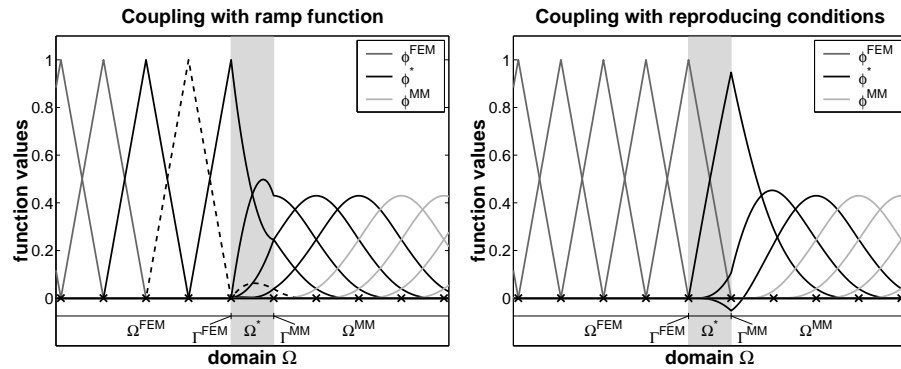


Figure 16: Resulting shape functions of the coupling approach with ramp function (subsection 6.3.1) and with reproducing conditions (subsection 6.3.2).

The resulting coupled set of shape functions in one dimension is depicted in the left part of Fig. 16; they build a PU of first order. The derivatives are discontinuous along Γ^{MM} and Γ^{FEM} as well as along interior interface element boundaries. These discontinuities do not adversely affect the overall results since they only affect a small number of nodes [20]. However, the higher rates of convergence of MMs can in general not be reproduced because the errors from the FEs dominate [20].

A very similar coupling approach in the MLPG framework may be found in [32] and [92], in the latter also a coupling to Boundary Element Methods is considered.

6.3.2 Coupling with Reproducing Conditions

Coupling with reproducing conditions was introduced by Huerta *et al.* in [64, 65]. Compared to coupling with ramp functions, this approach has the important advantage that a coupled PU with consistency of any desired order may be constructed, whereas the ramp function only achieves first order consistent PUs. The same discretization of the domain into areas Ω^{FEM} , Ω^* and Ω^{MM} and boundaries Γ^{FEM} and Γ^{MM} as described in subsection 6.3.1 and Fig. 15 is assumed.

An important difference of this approach is that the mesh-based approximation with FE shape functions is only complete in Ω^{FEM} and *not* in Ω^* . In Ω^* , only the FE shape functions of the nodes along Γ^{FEM} remain and are left unchanged throughout the coupling procedure; there are no FE shape functions of the nodes along Γ^{MM} , these nodes may be considered deactivated FEM nodes. Meshfree shape functions are constructed —e.g. with the MLS technique— for the nodes in $\Omega^{\text{MM}} \cup \Omega^* \setminus \Gamma^{\text{FEM}}$.

In this approach shape functions in Ω^{FEM} are provided by FEM shape functions only and in Ω^{MM} by meshfree techniques only. A coupling of the shape functions takes only place in Ω^* . There we write for the mixed approximation

$$\begin{aligned} u(\mathbf{x}) &\simeq u^{\text{MM}}(\mathbf{x}) + u^{\text{FEM}}(\mathbf{x}) \\ &\simeq \sum_{i \in I^{\text{MM}}} \Phi_i(\mathbf{x}) u(\mathbf{x}_i) + \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) u(\mathbf{x}_i). \end{aligned}$$

The objective now is to develop a mixed functional interpolation, with the desired consistency in Ω^* , *without any modification of the FE shape functions* [64, 65]. Thus, we want to deduce how to modify the meshfree approximation functions Φ_i in the presence of the (incomplete) FE shape functions. In section 4 many ways have been shown how to find an arbitrary order consistent meshfree approximation, e.g. via the MLS idea, Taylor series expansion etc. Here, we can employ the same ideas with the modified total approximation, which is $\sum_{i \in I^{\text{MM}}} \Phi_i(\mathbf{x}) u(\mathbf{x}_i) + \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) u(\mathbf{x}_i)$ instead of just $\sum_{i \in I^{\text{MM}}} \Phi_i(\mathbf{x}) u(\mathbf{x}_i)$.

In the following we do not always separate between sum over meshfree nodes $\sum_{i \in I^{\text{MM}}}$ and sum over mesh-based nodes $\sum_{i \in I^{\text{FEM}}}$, but just write $\sum_{i=1}^N$, where N is the total number of the nodes. This can be assumed without loss of generality, when $N_i = 0$ for $i \notin I^{\text{FEM}}$ and $\Phi_i = 0$ for $i \notin I^{\text{MM}}$. Here, the modified meshfree shape functions are deduced via the Taylor series expansion fully equivalent to subsection 4.2.2 where this is shown in detail. At this point

only the most important steps are shown:

$$\begin{aligned}
u^h(\mathbf{x}) &= \sum_{i \in I^{\text{MM}}} \Phi_i(\mathbf{x}) u(\mathbf{x}_i) + \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) u(\mathbf{x}_i) \\
&= \sum_{i=1}^N (\Phi_i(\mathbf{x}) + N_i(\mathbf{x})) u(\mathbf{x}_i) \\
&= \sum_{i=1}^N \left[(\mathbf{p}^T(\mathbf{x}_i - \mathbf{x}) \mathbf{a}(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) + N_i(\mathbf{x})) \cdot \right. \\
&\quad \left. \left(\sum_{|\alpha|=0}^{\infty} \frac{(\mathbf{x}_i - \mathbf{x})^\alpha}{|\alpha|!} D^\alpha u(\mathbf{x}) \right) \right] \\
&= \sum_{i=1}^N \left[\left((\mathbf{x}_i - \mathbf{x})^{\alpha^1} a_1(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) + (\mathbf{x}_i - \mathbf{x})^{\alpha^2} a_2(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) + \right. \right. \\
&\quad \left. \left. \dots + (\mathbf{x}_i - \mathbf{x})^{\alpha^k} a_k(\mathbf{x}) w(\mathbf{x} - \mathbf{x}_i) + N_i(\mathbf{x}) \right) \right. \\
&\quad \left. \left(\frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^1}}{|\alpha^1|!} D^{\alpha^1} u(\mathbf{x}) + \dots + \frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^k}}{|\alpha^k|!} D^{\alpha^k} u(\mathbf{x}) + \dots \right) \right].
\end{aligned}$$

Comparing the coefficients leads to the following system of equations:

$$\begin{aligned}
\sum_{i=1}^N \left((\mathbf{x}_i - \mathbf{x})^{\alpha^1} a_1 w(\mathbf{x} - \mathbf{x}_i) + \dots + (\mathbf{x}_i - \mathbf{x})^{\alpha^k} a_k w(\mathbf{x} - \mathbf{x}_i) + N_i(\mathbf{x}) \right) \frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^1}}{|\alpha^1|!} &= 1 \\
\sum_{i=1}^N \left((\mathbf{x}_i - \mathbf{x})^{\alpha^1} a_1 w(\mathbf{x} - \mathbf{x}_i) + \dots + (\mathbf{x}_i - \mathbf{x})^{\alpha^k} a_k w(\mathbf{x} - \mathbf{x}_i) + N_i(\mathbf{x}) \right) \frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^2}}{|\alpha^2|!} &= 0 \\
&\vdots \\
\sum_{i=1}^N \left((\mathbf{x}_i - \mathbf{x})^{\alpha^1} a_1 w(\mathbf{x} - \mathbf{x}_i) + \dots + (\mathbf{x}_i - \mathbf{x})^{\alpha^k} a_k w(\mathbf{x} - \mathbf{x}_i) + N_i(\mathbf{x}) \right) \frac{(\mathbf{x}_i - \mathbf{x})^{\alpha^k}}{|\alpha^k|!} &= 0.
\end{aligned}$$

Rearranging this gives in matrix notation

$$\begin{aligned}
\sum_{i \in I^{\text{MM}}} w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i - \mathbf{x}) \mathbf{p}^T(\mathbf{x}_i - \mathbf{x}) \mathbf{a}(\mathbf{x}) &= \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} - \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i - \mathbf{x}) \\
&= \mathbf{p}(0) - \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i - \mathbf{x}).
\end{aligned}$$

Thus, one can see that the difference to a meshfree-only approximation is a modified right hand side of the $k \times k$ systems of equations. Solving this for $\mathbf{a}(\mathbf{x})$ and inserting this into the approximation finally gives

$$\begin{aligned}
u^h(\mathbf{x}) &= \sum_{i \in I^{\text{MM}}} \mathbf{p}^T(\mathbf{x}_i - \mathbf{x}) \left[\sum_{i \in I^{\text{MM}}} w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i - \mathbf{x}) \mathbf{p}^T(\mathbf{x}_i - \mathbf{x}) \right]^{-1} \\
&\quad \left(\mathbf{p}(0) - \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i - \mathbf{x}) \right) w(\mathbf{x} - \mathbf{x}_i) u(\mathbf{x}_i) \\
&\quad + \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) u(\mathbf{x}_i).
\end{aligned}$$

Rearranging and applying the shifting procedure of the basis argument with $+\mathbf{x}$ gives

$$\begin{aligned}
u^h(\mathbf{x}) &= \sum_{i \in I^{\text{MM}}} \left(\mathbf{p}^T(\mathbf{x}) - \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) \mathbf{p}^T(\mathbf{x}_i) \right) \left[\sum_{i \in I^{\text{MM}}} w(\mathbf{x} - \mathbf{x}_i) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i) \right]^{-1} \\
&\quad \mathbf{p}(\mathbf{x}_i) w(\mathbf{x} - \mathbf{x}_i) u(\mathbf{x}_i) + \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) u(\mathbf{x}_i) \\
&= \sum_{i \in I^{\text{MM}}} \left(\mathbf{p}^T(\mathbf{x}) - \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) \mathbf{p}^T(\mathbf{x}_i) \right) [\mathbf{M}(\mathbf{x})]^{-1} \mathbf{p}(\mathbf{x}_i) w(\mathbf{x} - \mathbf{x}_i) u(\mathbf{x}_i) \\
&\quad + \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) u(\mathbf{x}_i),
\end{aligned}$$

where the moment matrix $\mathbf{M}(\mathbf{x})$ remains unchanged from the previous definition in section 4. The resulting coupled set of shape functions in one dimension is shown in the right part of Fig. 16. The shape functions Φ_i in the transition area are hierarchical [64, 65], because for any node \mathbf{x}_k the right

hand side of the system of equations becomes zero which can be seen easily: $\mathbf{p}(0) - \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}_k) \mathbf{p}(\mathbf{x}_i - \mathbf{x}_k) = \mathbf{p}(0) - \sum_{i \in I^{\text{FEM}}} \delta_{ik} \mathbf{p}(\mathbf{x}_i - \mathbf{x}_k) = \mathbf{p}(0) - \mathbf{p}(\mathbf{x}_k - \mathbf{x}_k) = 0$.

Concerning the continuity of the coupled approximation it is found [64, 65] that the coupled shape functions are continuous if, first, the same order of consistency is imposed all over Ω (i.e. for both FEs and particles), namely, $n^{\text{MM}} = n^{\text{FEM}}$. And second, the supports of the particles I^{MM} coincides exactly with the region where FEs do not have a complete basis. That is, no particles are added in “complete” FEs (i.e. elements where no node has been suppressed). Moreover, weighting functions are chopped off in those “complete” FEs.

It shall be mentioned that the above procedure can also be used to *enrich* the FE approximation with particle methods. For example, the following adaptive process seems attractive: compute an approximation with a coarse FE mesh, do an *a posteriori* error estimation and improve the solution with particles without any remeshing process [65].

In cases of finite element enrichment with MMs we consider $\Omega = \Omega^{\text{FEM}}$. Consequently, there is a complete FE basis in the entire domain Ω and only in a reduced area $\hat{\Omega}$ particles are added to improve the interpolation without removing the original complete FE interpolation. The resulting particle shape functions Φ_j are hierarchical and not linearly independent [65]. Thus, if every interpolation function is used, the stiffness matrix would be singular. To avoid this, once the enriching shape functions are evaluated, some of these interpolation functions are eliminated. Then, the stiffness matrix remains regular. In general, it is necessary to suppress a Φ_j (i.e. a particle) of the interpolation set for each polynomial in $\mathbf{p}(\mathbf{x})$ that FEs are able to capture exactly [65].

In the enriched region $\hat{\Omega}$, the consistency of the mixed interpolation n^{MM} must be larger than the order of the FE interpolation n^{FEM} because otherwise ($n^{\text{MM}} \leq n^{\text{FEM}}$) it would lead to particle shape functions which are zero everywhere [65]. Thus the basis of the particle shape functions must include at least one polynomial not reproducible by the FE interpolation; that means $n^{\text{MM}} > n^{\text{FEM}}$ is necessary. Changing the order of consistency induces discontinuities in the approximation along the enriched boundary [65]. However, if the boundary $\partial\hat{\Omega}$ coincides with an area where FEs capture accurately the solution, those discontinuities due to the enrichment are going to be small.

6.3.3 Bridging Scale Method

This bridging scale method has been introduced for coupling in the RKPM context in [100], also discussed in [129]. Starting point is an incomplete fi-

nite element bases $\{N_i(\mathbf{x})\}_{i \in I^{\text{FE}}}$ and a complete meshfree interpolation *on the whole domain* $\{\Phi_i(\mathbf{x})\}_{i \in I^{\text{MM}}}$, which is in contrast to the approach of Huerta, see subsection 6.3.2, where particles may only be introduced where the FE interpolation is incomplete. For the bridging scale method the necessity to evaluate meshfree shape functions in the whole domain obviously alleviates an important argument for coupling, which is the reduction of computing time.

Firstly, the viewpoint taken in the bridging scale method shall be briefly outlined. One wants to hierarchically decompose a function $u(\mathbf{x})$ based on some projection operator \mathcal{P} as

$$u = \mathcal{P}u + w - \mathcal{P}w, \quad (6.2)$$

where w is some enrichment function. The total enrichment term $w - \mathcal{P}w$ contains only the part of u which is not representable by the projection [129]. The term $\mathcal{P}w$ is called bridging scale term and allows greater freedom in the choice of w , because without this term, w must be chosen to be a function whose projection is zero. For the projection of u onto the finite element basis we write

$$\mathcal{P}u(\mathbf{x}) = \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) u(\mathbf{x}_i).$$

The meshfree interpolation $w(\mathbf{x}) = \sum_{i \in I^{\text{MM}}} \Phi_i(\mathbf{x}) u(\mathbf{x}_i)$ is thought of as an enrichment of this FE basis. For the projection $\mathcal{P}w$ of the meshfree interpolation w onto the finite element basis follows analogously

$$\begin{aligned} \mathcal{P}w^h(\mathbf{x}) &= \sum_{j \in I^{\text{FEM}}} N_j(\mathbf{x}) w(\mathbf{x}), \\ &= \sum_{j \in I^{\text{FEM}}} N_j(\mathbf{x}) \sum_{i \in I^{\text{MM}}} \Phi_i(\mathbf{x}_j) u(\mathbf{x}_i). \end{aligned}$$

Inserting now these definitions of $\mathcal{P}u$, w and $\mathcal{P}w$ into equation 6.2 gives

$$u(\mathbf{x}) = \sum_{i \in I^{\text{FEM}}} N_i(\mathbf{x}) u(\mathbf{x}_i) + \sum_{i \in I^{\text{MM}}} \left[\Phi_i(\mathbf{x}) - \sum_{j \in I^{\text{FEM}}} N_j(\mathbf{x}) \Phi_i(\mathbf{x}_j) \right] u(\mathbf{x}_i), \quad (6.3)$$

where the modified meshfree shape functions may immediately be extracted.

For a consistency proof of this formulation see [129]. In [66] the bridging scale method has been compared with the coupling approach of Huerta [65], see subsection 6.3.2. An important difference is that the term $\Phi_i(\mathbf{x}_j)$ in equation 6.3 is constant, whereas for the coupling approach of Huerta it may be shown

after modifications of the structure of the resulting expressions that the analogous term is a *function* of \mathbf{x} [66]. Furthermore, in the bridging scale method—in order to ensure the continuity of the approximation—particles for the construction of the meshfree basis have to cover the whole domain. In contrast, in the approach of Huerta particles are only needed in areas, where the FEM shape functions are not complete. The continuity consideration is directly related to problems of the bridging scale method for the imposition of EBCs: The resulting meshfree shape functions are only zero at the finite element nodes, however, not along element edges/faces along the boundary in 2D/3D respectively. Therefore, it is not surprising that the approach of Huerta turns out to be superior in [66].

6.3.4 Coupling with Lagrangian Multipliers

The coupling approach with Lagrangian multipliers couples two distinct domains, one for the FE-part and the other for the MM-part via the weak form [63]. Consequently, this approach is very different to the previously discussed approaches, no coupled shape functions with a certain order of consistency are developed. As for a purely MM approach, it was shown that the rates of convergence for a combination of MMs and FE can exceed those of the FEM. This method shares all the disadvantages as mentioned for the impositions of EBCs with Lagrangian multipliers, see subsection 6.1.1.

6.4 Discontinuities

The continuity of meshfree shape functions is often considerably higher than FEM shape functions. In fact, they can be built with any desired order of continuity depending most importantly on the choice of the weight function. The resulting derivatives of meshfree interpolations are also smooth leading in general to very desirable properties, like smooth stresses etc. However, many practical problems involve physically justified discontinuities. For example, in crack simulation the displacement field is discontinuous, whereas in a structural analysis of two different connected materials the stresses are discontinuous; in the prior case the discontinuity is related to the interpolation itself, in the latter case only to the derivatives (discontinuous derivatives occur whenever the coefficients of the PDE under consideration are discontinuous).

MMs need certain techniques to handle these discontinuities. Classical mesh-based methods have problems to handle these problems, because there the dis-

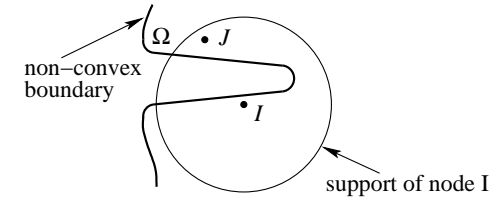


Figure 17: One has to be careful for non-convex boundaries. The support of node I should be modified, therefore, the same methods as for discontinuity treatment may be used.

continuity must align with element edges; although also for these methods ways have been found to overcome this problem (e.g. [19]).

It should be mentioned that the treatment of discontinuities has similar features than the treatment of non-convex boundaries, see Fig. 17. We cite from [63]:

One has to be careful with performing MLS for a domain which is strongly non-convex. Here, one can think of a domain with a sharp concave corner. To achieve that MLS is well defined for such a domain and to have that the shape functions are continuous on the domain, it is possible that shape functions become non-zero on parts of the domain (think of the opposite side of the corner) where it is more likely that they are zero. Hence, nodal points can influence the approximant u^h on parts of the domain where it is not really convenient to have this influence.

Here, we divide methods which modify the supports along the discontinuity, see subsection 6.4.1 to 6.4.3, and thus which incorporate discontinuous approximations as an enrichment of their basis functions, see subsection 6.4.4. See [116] for an interesting comparison of the methods which modify the supports.

6.4.1 Visibility Criterion

The visibility criterion, introduced in [18], may be easily understood by considering the discontinuity opaque for “rays of light” coming from the nodes. That is, for the modification of a support of node I one considers light coming from

the coordinates of node I and truncates the part of the support which is in the shadow of the discontinuity. This is depicted in Fig. 18.

A major problem of this approach is that at the discontinuity tips an artificial discontinuity inside the domain is constructed and the resulting shape functions are consequently not even C^0 continuous. Convergence may still be reached [78], however, significant errors result and oscillations around the tip can occur especially for larger dilatation parameters [116]. The methods discussed in the following may be considered as fixed versions of the short comings of the visibility criterion and show differences only at the treatment around the discontinuity tips.

It shall further be mentioned that for all methods that modify the support—which in fact is somehow a reduction of the prior size—there may be problems in the regularity of the $k \times k$ system of equations, see subsection 4.6, because less supports overlap with the modified support. Therefore, it may be necessary to increase the support size leading to a larger band width of the resulting system of equations. This aspect has been pointed out in [21].

6.4.2 Diffraction Method

The diffraction method [16, 116] considers the diffraction of the rays around the tip of the discontinuity. For the evaluation of the weighting function at a certain evaluation point (usually an integration point) the input parameter of $w(\|\mathbf{x} - \mathbf{x}_I\|) = w(d_I)$ is changed in the following way: Define $s_0 = \|\mathbf{x} - \mathbf{x}_I\|$, s_1 being the distance from the node to the crack tip, $s_1 = \|\mathbf{x}_c - \mathbf{x}_I\|$, and s_2 the distance from the crack tip to the evaluation point, $s_2 = \|\mathbf{x} - \mathbf{x}_c\|$. Then we change d_I as [116]

$$d_I = \left(\frac{s_1 + s_2}{s_0} \right)^\gamma s_0;$$

in [16] only $\gamma = 1$, i.e. $d_I = s_1 + s_2 = \|\mathbf{x}_c - \mathbf{x}_I\| + \|\mathbf{x} - \mathbf{x}_I\|$, has been proposed. Reasonable choices for γ are 1 or 2 [116], however, optimal values for γ are not available and problem specific. The derivatives of the resulting shape function is not continuous directly at the crack tip, however, this poses no difficulties as long as no integration point is placed there [116].

The modification of the support according to the diffraction method may be seen in Fig. 18. A natural extension of the diffraction method for the case of multiple discontinuities per support may be found in [110].

6.4.3 Transparency Method

In [116] the transparency method is introduced. Here, the function is smoothed around a discontinuity by endowing the surface of the discontinuity with a varying degree of transparency. The tip of the discontinuity is considered completely transparent and becomes more and more opaque with increasing distance from the tip. For the modification of the input parameter of the weighting function d_I follows:

$$d_I = s_0 + \rho_I \left(\frac{s_c}{\bar{s}_c} \right)^\gamma, \gamma \geq 2,$$

where $s_0 = \|\mathbf{x} - \mathbf{x}_I\|$, ρ_I is the dilatation parameter of node I , s_c is the intersection of the line $\overline{\mathbf{x}\mathbf{x}_I}$ with the discontinuity and \bar{s}_c is the distance from the crack tip where the discontinuity is completely opaque. For nodes directly adjacent to the discontinuity a special treatment is proposed [116]. The value γ of this approach is also a free value which has to be adjusted with empirical arguments. The resulting derivatives are continuous also at the crack tip.

6.4.4 PUM Ideas

Belytschko *et al.* propose in [21] a discontinuous enrichment of the approximation by means of including a jump function along the discontinuity and a specific solution at the discontinuity tip in the extrinsic basis. Consequently, this method can be considered a member of the PUMs, see subsection 5.7. For similar approaches see also [19, 77].

6.5 h -Adaptivity

In adaptive simulations nodes are added and removed over subsequent iteration steps. The aim is to achieve a prescribed accuracy with minimal number of nodes or to capture a local behavior in an optimal way. Mesh-based methods such as the FEM require a permanent update of the connectivity relations, i.e. a conforming mesh must be maintained. Automatic remeshing routines, however, may fail in complex geometric situations especially in three dimensions; these aspects have already been mentioned in subsection 5.10. In contrast, MMs seem ideally suited for adaptive procedures as they naturally compute the node connectivity at runtime.

Most adaptive algorithms of MMs proceed as follows:

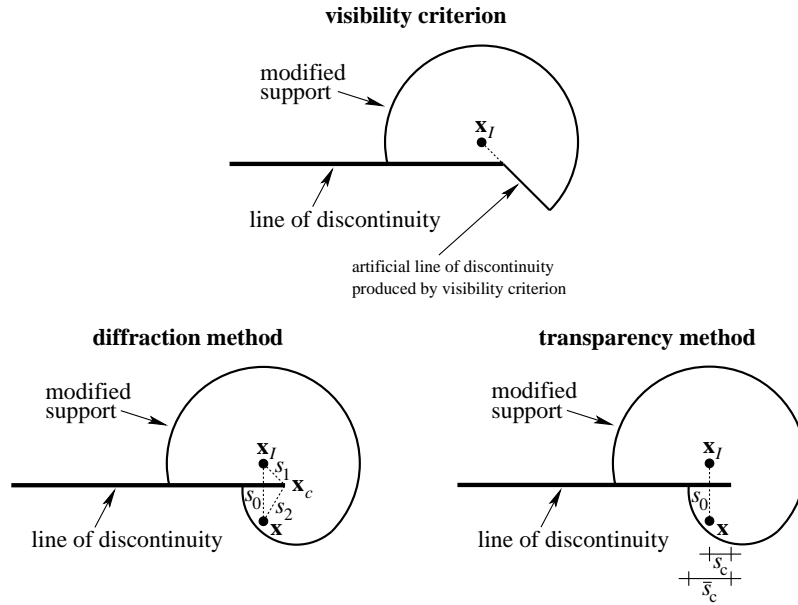


Figure 18: Visibility criterion, diffraction and transparency method for the treatment of discontinuities.

- Error indication/estimation: In this step the error of a previous iteration (or time) step is estimated *a posteriori*. Regions of the domains are identified where the error is relatively large and refinement is most effectively. For a description of error indicators/estimators and related ideas in MMs, see e.g. [35, 50, 95, 105]. We do not go into further detail, because the principles of error indicators (residual-based, gradient-based, multi-scale decomposition etc.) are comparable to those of standard mesh-based methods.
- Construction of a (local) Voronoi diagram: A Voronoi diagram is constructed with respect to the current node distribution in the region identified by the error estimator.
- Insertion of particles: The Voronoi diagram is used as a geometrical reference for particle insertion, i.e. particles are added at the corners of the Voronoi diagram. Additionally, the Voronoi diagram may be used to build efficient data structures for the newly inserted nodes [101], for example simplifying neighbour-searching procedures.
- Adjustment of the dilatation parameters: Adding or removing particles without adjusting the support size of the shape functions by means of the dilatation parameter ρ leads in the prior case to a dramatic increase of the band width of the resulting system of equations and in the latter case to a possible loss of regularity of the $k \times k$ systems of equations of the MLS or RKPM method. The Voronoi diagram may as well be used to adjust the dilatation parameters. Lu *et al.* suggest in [101] to adjust ρ_I of a certain node I by building a set of neighbouring nodes I^α —those particles with neighbouring Voronoi cells of the current particle's Voronoi cell— and to compute

$$\rho_I = \alpha \cdot \max \{d_J : d_J = \overline{\mathbf{x}_I \mathbf{x}_J}, \forall \mathbf{x}_J \in I^\alpha\},$$

with $\alpha \approx 2.0$ in general. It should be mentioned that it is the advantage of mesh-based methods that the support size of the shape functions is directly determined by the mesh which is absent for MMs.

Also a stopping criteria is part of the adaptive algorithm. For example in structural analysis, it may be defined by means of the change of potential energy; if the relative change decreases under a certain limit the adaption is stopped [105]. It should be noted that a Voronoi diagram may be circumvented by other ideas for the introduction of new nodes, see e.g. [41] where new nodes are added around node I with respect to the distance of the nearest neighbour of node I .

A noteworthy idea for an adaptive procedure with coupled FEM/EFG approximation according to the coupling approach of Huerta [65], discussed in 6.3.2, has been showed in [45]. There, first an approximation is computed with a finite element mesh which is followed by an error estimation and finally, elements with large error are indicated, removed and replaced by particles, which may easily be refined in subsequent steps. Thereby, one makes very selectively profit of the advantageous properties of MMs in adaptive procedures.

6.6 Parallelization

In this subsection we follow Danielson, Hao, Liu, Uras and Li [36]. The parallelization is done with respect to the integration which needs significantly more computing time than classical mesh-based methods. This problem may be considered to be trivially parallelizable with a complexity of $O\left(\frac{N}{p}\right)$ where p is the number of processors and N the number of integration points [55]. The parallelization of the final system of equations is not considered here as this is not necessarily a MM specific problem, remarks for this problem may be found in [55].

The basic principle of parallel computing is to balance the computational load among processors while minimizing interprocessor communication (build partitions of “same size”, minimize partition interface). The first phase of a parallel computation is the partitioning phase. In MMs, integration points are distributed to processors and are uniquely defined there. To retain data locality, particles are redundantly defined (duplicated) on all processors possessing integration points contributing to these particles. Parallel SPH implementations typically partition the particles (which are identical to the integration points), whereas for Galerkin MMs partitioning is done on the integration points. Partitioning procedures may be based on

- graph theory: Each integration point has a list of support particles within its DOI that it must provide a contribution. Integration points in MMs typically contribute to many more nodes than those of similar FE models do, thus, the graphs can be very large with many edges. Reduction of the graph is possible by considering only the nearest particles (geometrical criteria). Also the reduced graph results into nearly perfect separated partitions. In all cases, no significant reduction in performance occurred by using the reduced graph partitioning instead of the full graph partitioning.
- geometric technique: They only require the integration point coordinates for partitioning. Groups of integration points are built in the same spatial

proximity. Methods are Recursive Coordinate Bisection (RCB), Hilbert Space Filling Curve (HSFC) and Unbalanced RCB. It is claimed in [36] that the results are not always satisfactory, as imbalances up to 20% occurred. However, in [55] the space filling curve approach has been applied successfully.

The second phase is the parallel analysis phase, where the problem is calculated. The basic parallel approach is for each processor to perform an analysis on its own partition. These are performed as if they were separate analysis, except that the contributions to the force vectors are sent to and from other processors for the duplicated nodes. MPI statements are used with non-blocking communication (MPI ISEND/Irecv) to avoid possible deadlocks and overhead associated with buffering.

When the analysis is finally completed, in the third phase, separate software is run serially to gather the individual processor output files into a single coherent data structure for post-processing.

With this procedure significant speedup could be gained. The authors of [36] claim that the enforcement of EBCs in parallel merits more investigation.

6.7 Solving the Global System of Equations

In this subsection the solution of the *total* system of equations —and not the $k \times k$ systems of the MLS/RKPM procedure— shall be briefly discussed. The regularity of the $k \times k$ matrix inversions of the MLS or RKPM for the construction of a PU does not ensure the regularity of the global system of equations, hence the solvability of the discrete variational problem. The matrix for the global problem may for example be singular if a numerical integration rule for the weak form is employed which is not accurate enough.

Moreover, in enrichment cases, the shape functions of the particles are not always linearly independent, unless they are treated in a certain way (e.g. eliminating several interpolation functions) [65]. Additionally, the global matrix may be ill-conditioned for various distributions of particle points, when the particle distributions led already to ill-conditioned $k \times k$ matrices. These topics have already been discussed in subsections 4.4, 4.6 and 6.2 respectively and are not repeated here.

When using MMs with intrinsic basis only, the size of the global matrix is the same for any consistency. For a large value of consistency order n , the increased amount of computational work lies only in the solution of the $k \times k$

MLS/RKPM-systems at every evaluation point. The expense for solving the global system is basically the same for any degree n [60].

In MMs the sparsity of the matrix differs from mesh-based methods. As there are more nodes in the support of a particle, the matrix is less sparse. This influences the behaviour of iterative solvers. In [83] the convergence and timing performance of some well-known Krylov subspace methods on solving linear systems from meshfree discretizations are examined. Krylov subspace techniques are among the most important iterative techniques available for solving large linear systems and some methods such as CGS, BCG, GMRES, CGNE are tested. It is observed that the BCG and CGS exhibit slightly faster convergence rates, but they also show very irregular behaviours (oscillations in the residual-iteration diagram) [83]. The GMRES displays a very slow convergence rate, but does not show oscillatory behaviour [83].

A multilevel approach for the solution of elliptic PDEs with MMs has been proposed in [54].

6.8 Summary and Conclusion

In this section we discussed some problems which frequently occur in MMs and gave reason for various publications.

Many different ideas have been introduced for the handling of essential boundary conditions in MMs. The lack of the Kronecker delta property in meshfree methods makes this topic important, in contrast to mesh-based methods where EBCs can be imposed trivially. Methods have been shown which e.g. work with modified weak forms or coupled/manipulated shape functions achieving Kronecker delta property at the boundary. Advantages and disadvantages have already been mentioned previously in each of the subsections.

Integration has been intensively discussed in subsection 6.2. Integration is the most time-consuming part in a MM calculation due to the large number of integration points needed for a sufficiently accurate evaluation of the integrals in the weak form. In collocation MMs the weak form reduces to the strong form and integration is not needed which is their main advantage. Galerkin MMs with nodal integration are closely related to collocation methods. Accuracy and stability are the weak points of both collocation MMs and Galerkin MMs with nodal integration.

The accuracy of full integration compared to nodal integration is considerably higher, see e.g. [15]. Integration with background mesh or cell structure has been proposed for the earlier MMs. The method can then be considered

as pseudo-meshfree and the major advantage of MMs to approximate solutions without a mesh is alleviated. Therefore, integration over supports seems attractive as it is truly meshfree. The problem of misalignment of the integration domain and the support of the integrands leads to more elaborate integration ideas like integration over intersections of supports. Then integration domain and support of the integrand align and the accuracy is superior, however, the computational burden associated with this idea is often higher and is a serious handicap.

In subsection 6.3 several methods have been discussed for the coupling of mesh-based methods with meshfree methods. The aim is always to combine the advantages of each method, above all the computational efficiency of mesh-based with the unnecessary of maintaining a mesh in MMs. Still today MMs turn out to be used only in rather special applications, like crack growth, which is due to their time-consuming integration. For engineering applications like flow simulation, structural dynamics etc. it seems not probable that MMs are used for the simulation in the whole domain as a *standard* tool. It is our belief that only in combination with mesh-based methods we can expect to use MMs with practical relevance in these kind of problems. Consider e.g. a problem where in a heart-valve simulation the arteries itself is modeled with FEM while MMs are used only in the valve region where a mesh is of significant disadvantage due to the large geometrical changes of the computational domain. In our opinion, the coupling of mesh-based and meshfree methods is an essential aspect for the success of MMs in practical use.

Methods for the treatment of discontinuities in MMs are discussed in 6.4, separating the approaches into two different principles: Thus which modify the supports of the weighting functions (and resulting shape functions) and thus that use PUM ideas (see subsection 6.1.8) to enrich the approximation basis in order to reflect the special solution characteristics around a discontinuity.

In subsection 6.6 and 6.7 some statements are given for the parallelization and the solution of the global system of equations in MMs.

7 Conclusion

In this paper an overview and a classification of meshfree methods has been presented. The similarities and differences between the variety of meshfree methods have been pointed out.

Concepts for a construction of a PU are explained with emphasis on the MLS and RKPM. It turned out that MLS and RKPM are very similar but not

identical (see subsection 4.3). Often it does not make sense to overemphasize the aspect of a method to be based on MLS or RKPM, especially if the slight difference between the MLS and RKPM is not pointed out or used. One should keep in mind that all the MMs in section 5 both work based on MLS *and* RKPM, but there is a certain difference between these concepts.

MLS and RKPM are separated from the resulting MMs themselves. It was shown that in case of approximations with intrinsic basis only, the PU functions are directly the shape functions, and thus the separation might be called somewhat superfluous in this case. However, to base our classification on exclusive properties, we believe that constructing a PU and choosing an approximation are *two* steps in general. This is obvious for cases of approximations with an additional extrinsic basis.

The MMs themselves have been explained in detail, taking into account the different viewpoints and origins of each method. Often, we have focused on pointing out important characteristic features rather than on explaining how the method functions. The latter becomes already clear from sections 3 and 4. We found that SPH and DEM are problematical choices for a MM, the former due to the lack of consistency and the latter due to the negligence of some derivative terms. A number of corrected versions of the SPH exists which fix this lack of consistency, and the EFG method may also be viewed as a fixed version of the DEM. SPH and EFG may be the most popular MMs in practice. The first is a representative of the collocation MMs which do not require a time-consuming integration but may show accuracy and stability problems; the latter is a representative of the Galerkin MMs which solve the weak form of a problem with a comparably high accuracy in general, however, requiring an expansive integration. MMs with an extrinsic basis are representatives of the PUM idea; the GFEM, XFEM, PUFEM etc. fall into this class, too. The LBIE is the meshfree equivalent of the boundary element methods and may be used efficiently for problems where fundamental solutions are known. Some MMs which are not based on the MLS/RKPM principle, like the NEM and MFEM have also been discussed. However, it was not possible to present a complete description of *all* available MMs.

This paper also discusses intensively problems which are related to MMs. The disadvantage of MMs not to be interpolating in general makes the imposition of EBCs awkward and many techniques have been developed for this purpose. Procedures for the integration of the weak form in Galerkin MMs have been shown. Coupling meshfree and mesh-based methods is a very promising way as advantages of each method can be used where they are needed. Aspects of discontinuity treatment, parallelization and solvers have also been discussed.

We hope this paper to be a helpful tool for the reader's successful work with Meshfree Methods.

References

- [1] Aluru, N.R.: A point collocation method based on reproducing kernel approximations. *Internat. J. Numer. Methods Engrg.*, **47**, 1083 – 1121, 2000.
- [2] Atluri, S.N.; Cho, J.Y.; Kim, H.-G.: Analysis of Thin Beams, Using the Meshless Local Petrov-Galerkin Method, with Generalized Moving Least Squares Interpolations. *Comput. Mech.*, **24**, 334 – 347, 1999.
- [3] Atluri, S.N.; Kim, H.-G.; Cho, J.Y.: A Critical Assessment of the Truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) Methods. *Comput. Mech.*, **24**, 348 – 372, 1999.
- [4] Atluri, S.N.; Shen, S.: *The Meshless Local Petrov-Galerkin (MLPG) Method*. Tech Science Press, Stuttgart, 2002.
- [5] Atluri, S.N.; Sladek, J.; Sladek, V.; Zhu, T.: The Local Boundary Integral Equation (LBIE) and its Meshless Implementation for Linear Elasticity. *Comput. Mech.*, **25**, 180 – 198, 2000.
- [6] Atluri, S.N.; Zhu, T.: A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics. *Comput. Mech.*, **22**, 117 – 127, 1998.
- [7] Atluri, S.N.; Zhu, T.: New concepts in meshless methods. *Internat. J. Numer. Methods Engrg.*, **47**, 537 – 556, 2000.
- [8] Babuška, I.; Banerjee, U.; Osborn, J.E.: Meshless and generalized finite element methods: A survey of some major results. In *Meshfree Methods for Partial Differential Equations*. (Griebel, M.; Schweitzer, M.A., Eds.), Vol. 26, Springer Verlag, Berlin, 2002.
- [9] Babuška, I.; Banerjee, U.; Osborn, J.E.: Survey of meshless and generalized finite element methods: A unified approach. Technical Report 02-40, TICAM, The University of Texas at Austin, 2002.
- [10] Babuška, I.; Melenk, J.M.: The partition of unity finite element method. Technical Report BN-1185, Institute for Physical Science and Technology, University of Maryland, 1995.

- [11] Babuška, I.; Melenk, J.M.: The Partition of Unity Method. *Internat. J. Numer. Methods Engrg.*, **40**, 727 – 758, 1997.
- [12] Beissel, S.; Belytschko, T.: Nodal integration of the element-free Galerkin method. *Comp. Methods Appl. Mech. Engrg.*, **139**, 49 – 74, 1996.
- [13] Belikov, V.V.; Ivanov, V.D.; Kontorovich, V.K.; Korytnik, S.A.; Semenov, A.Y.: The non-Sibsonian interpolation: a new method of interpolation of the values of a function on an arbitrary set of points. *Comp. Math. Math. Phys.*, **37**, 9 – 15, 1997.
- [14] Belytschko, T.; Guo, Y.; Liu, W.K.; Xiao, S.P.: A unified stability analysis of meshless particle methods. *Internat. J. Numer. Methods Engrg.*, **48**, 1359 – 1400, 2000.
- [15] Belytschko, T.; Krongauz, Y.; Dolbow, J.; Gerlach, C.: On the completeness of meshfree particle methods. *Internat. J. Numer. Methods Engrg.*, **43**, 785 – 819, 1998.
- [16] Belytschko, T.; Krongauz, Y.; Fleming, M.; Organ, D.; Liu, W.K.S.: Smoothing and accelerated computations in the element free Galerkin method. *J. Comput. Appl. Math.*, **74**, 111 – 126, 1996.
- [17] Belytschko, T.; Krongauz, Y.; Organ, D.; Fleming, M.; Krysl, P.: Meshless Methods: An Overview and Recent Developments. *Comp. Methods Appl. Mech. Engrg.*, **139**, 3 – 47, 1996.
- [18] Belytschko, T.; Lu, Y.Y.; Gu, L.: Element-free Galerkin Methods. *Internat. J. Numer. Methods Engrg.*, **37**, 229 – 256, 1994.
- [19] Belytschko, T.; Moës, N.; Usui, S.; Parimi, C.: Arbitrary discontinuities in finite elements. *Internat. J. Numer. Methods Engrg.*, **50**, 993 – 1013, 2001.
- [20] Belytschko, T.; Organ, D.; Krongauz, Y.: A Coupled Finite Element–Element-free Galerkin Method. *Comput. Mech.*, **17**, 186 – 195, 1995.
- [21] Belytschko, T.; Ventura, G.; Xu, J.X.: New methods for discontinuity and crack modeling in EFG. In *Meshfree Methods for Partial Differential Equations*. (Griebel, M.; Schweitzer, M.A., Eds.), Vol. 26, Springer Verlag, Berlin, 2002.
- [22] Bonet, J.; Kulasegaram, S.: Correction and Stabilization of Smooth Particle Hydrodynamics Methods with Applications in Metal Forming Simulations. *Internat. J. Numer. Methods Engrg.*, **47**, 1189 – 1214, 2000.

- [23] Braun, J.; Sambridge, M.: A numerical method for solving partial differential equations on highly irregular evolving grids. *Nature*, **376**, 655 – 660, 1995.
- [24] Brezzi, F.: On the existence, uniqueness and approximation of saddle-point problems arising from Lagrange multipliers. *RAIRO Anal. Numér.*, **R-2**, 129 – 151, 1974.
- [25] Chen, J.-S.; Pan, C.; Wu, C.-I.: Large Deformation Analysis of Rubber based on a Reproducing Kernel Particle Method. *Comput. Mech.*, **19**, 211 – 227, 1997.
- [26] Chen, J.S.; Han, W.; You, Y.; Meng, X.: A reproducing kernel method with nodal interpolation property. *Internat. J. Numer. Methods Engrg.*, **56**, 935 – 960, 2003.
- [27] Chen, J.S.; Liu, W.K. (eds.): Meshless particle methods. *Comput. Mech.*, **25**(2-3, special issue), 99 – 317, 2000.
- [28] Chen, J.S.; Liu, W.K. (eds.): Meshless methods: Recent advances and new applications. *Comp. Methods Appl. Mech. Engrg.*, **193**(12-14, special issue), 933 – 1321, 2004.
- [29] Chen, J.S.; Wang, H.-P.: New Boundary Condition Treatments in Mesh-free Computation of Contact Problems. *Comp. Methods Appl. Mech. Engrg.*, **187**, 441 – 468, 2000.
- [30] Chen, J.S.; Wu, C.T.; You, Y.: A Stabilized Conforming Nodal Integration for Galerkin Mesh-free Methods. *Internat. J. Numer. Methods Engrg.*, **50**, 435 – 466, 2001.
- [31] Chen, J.S.; Yoon, S.; H.P. Wang, W.K. Liu: An improved reproducing kernel particle method for nearly incompressible finite elasticity. *Comp. Methods Appl. Mech. Engrg.*, **181**, 117 – 145, 2000.
- [32] Chen, T.; Raju, I.S.: Coupling finite element and meshless local Petrov-Galerkin methods for two-dimensional potential problems. AIAA 2002-1659, NASA Langley Research Center, Hampton, USA, 2002.
- [33] Choe, H.J.; Kim, D.W.; Kim, H.H.; Kim, Y.: Meshless Method for the Stationary Incompressible Navier-Stokes Equations. *Discrete and Continuous Dynamical Systems - Series B*, **1**(4), 495 – 526, 2001.

- [34] Choi, Y.J.; Kim, S.J.: Node Generation Scheme for the MeshFree Method by Voronoi Diagram and Weighted Bubble Packing. *Fifth U.S. National Congress on Computational Mechanics*, Boulder, CO, 1999.
- [35] Chung, H.J.; Belytschko, T.: An error estimate in the EFG method. *Comput. Mech.*, **21**, 91 – 100, 1998.
- [36] Danielson, K.T.; Hao, S.; Liu, W.K.; Uras, R.A.; Li, S.: Parallel computation of meshless methods for explicit dynamic analysis. *Internat. J. Numer. Methods Engrg.*, **47**, 1323 – 1341, 2000.
- [37] De, S.; Bathe, K.J.: The Method of Finite Spheres. *Comput. Mech.*, **25**, 329 – 345, 2000.
- [38] Dilts, G.A.: Moving-Least-Squares-Particle Hydrodynamics –I. Consistency and Stability. *Internat. J. Numer. Methods Engrg.*, **44**, 1115 – 1155, 1999.
- [39] Dolbow, J.; Belytschko, T.: Numerical Integration of the Galerkin Weak Form in Meshfree Methods. *Comput. Mech.*, **23**, 219 – 230, 1999.
- [40] Duarte, C.A.: A review of some meshless methods to solve partial differential equations. Technical Report 95-06, TICAM, The University of Texas at Austin, 1995.
- [41] Duarte, C.A.; Oden, J.T.: An h-p adaptive method using clouds. *Comp. Methods Appl. Mech. Engrg.*, **139**, 237 – 262, 1996.
- [42] Duarte, C.A.M.; Oden, J.T.: Hp clouds – a meshless method to solve boundary-value problems. Technical Report 95-05, TICAM, The University of Texas at Austin, 1995.
- [43] Duarte, C.A.M.; Oden, J.T.: H-p clouds – an h-p meshless method. *Numer. Methods Partial Differential Equations*, **12**, 673 – 705, 1996.
- [44] Dyka, C.T.; Ingel, R.P.: An approach for tension instability in smoothed particle hydrodynamics. *Computers & Structures*, **57**, 573 – 580, 1995.
- [45] Fernández-Méndez, S.; Huerta, A.: Coupling finite elements and particles for adaptivity. In *Meshfree Methods for Partial Differential Equations*. (Griebel, M.; Schweitzer, M.A., Eds.), Vol. 26, Springer Verlag, Berlin, 2002.

- [46] Fernández-Méndez, S.; Huerta, A.: Imposing essential boundary conditions in mesh-free methods. *Comp. Methods Appl. Mech. Engrg.*, **193**, 1257 – 1275, 2004.
- [47] Fries, T.P.; Matthies, H.G.: A Review of Petrov-Galerkin Stabilization Approaches and an Extension to Meshfree Methods. Informatikbericht-Nr. 2004-01, Technical University of Braunschweig, (<http://opus.tu-bs.de/opus/volltexte/2004/549/>), Brunswick, 2004.
- [48] Fries, T.P.; Matthies, H.G.: Meshfree Petrov-Galerkin Methods for the Incompressible Navier-Stokes Equations. In *Meshfree Methods for Partial Differential Equations*. (Griebel, M.; Schweitzer, M.A., Eds.), Springer Verlag, Berlin, 2004 (to appear).
- [49] Garcia, O.; Fancello, E.A.; de Barcellos, C.S.; Duarte, C.A.: hp-Clouds in Mindlin's Thick Plate Model. *Internat. J. Numer. Methods Engrg.*, **47**, 1381 – 1400, 2000.
- [50] Gavete, L.; Cuesta, J.L.; Ruiz, A.: A procedure for approximation of the error in the EFG method. *Internat. J. Numer. Methods Engrg.*, **53**, 677 – 690, 2002.
- [51] Gerstner, T.; Griebel, M.: Numerical integration using sparse grids. *Numer. Algorithms*, **18**, 209 – 232, 1998.
- [52] Gingold, R.A.; Monaghan, J.J.: Kernel Estimates as a Basis for General Particle Methods in Hydrodynamics. *J. Comput. Phys.*, **46**, 429 – 453, 1982.
- [53] Griebel, M.; Schweitzer, M.A.: A particle-partition of unity method–Part II: Efficient cover construction and reliable integration. *SIAM J. Sci. Comput.*, **23**(5), 1655 – 1682, 2002.
- [54] Griebel, M.; Schweitzer, M.A.: A particle-partition of unity method–Part III: A multilevel solver. *SIAM J. Sci. Comput.*, **24**, 377 – 409, 2002.
- [55] Griebel, M.; Schweitzer, M.A.: A particle-partition of unity method–Part IV: Parallelization. In *Meshfree Methods for Partial Differential Equations*. (Griebel, M.; Schweitzer, M.A., Eds.), Vol. 26, Springer Verlag, Berlin, 2002.
- [56] Griebel, M.; Schweitzer, M.A. (eds.): *Meshfree Methods for Partial Differential Equations*, Vol. 26. Springer Verlag, Berlin, 2002.

- [57] Günther, F.C.: *A Meshfree Formulation for the Numerical Solution of the Viscous Compressible Navier-Stokes Equations*. Dissertation, Northwestern University, Evanston, IL, 1998.
- [58] Günther, F.C.; Liu, W.K.: Implementation of Boundary Conditions for Meshless Methods. *Comp. Methods Appl. Mech. Engrg.*, **163**, 205 – 230, 1998.
- [59] Han, W.; Meng, X.: Error analysis of the reproducing kernel particle method. *Comp. Methods Appl. Mech. Engrg.*, **190**, 6157 – 6181, 2001.
- [60] Han, W.; Meng, X.: Some studies of the reproducing kernel particle method. In *Meshfree Methods for Partial Differential Equations*. (Griebel, M.; Schweitzer, M.A., Eds.), Vol. 26, Springer Verlag, Berlin, 2002.
- [61] Hao, S.; Liu, W.K.: Revisit of Moving Particle Finite Element Method. *Proceedings of the Fifth World Congress on Computational Mechanics (WCCM V)*, Vienna, 2002.
- [62] Hao, S.; Park, H.S.; Liu, W.K.: Moving particle finite element method. *Internat. J. Numer. Methods Engrg.*, **53**, 1937 – 1958, 2002.
- [63] Hegen, D.: Element-free Galerkin Methods in Combination with Finite Element Approaches. *Comp. Methods Appl. Mech. Engrg.*, **135**, 143 – 166, 1996.
- [64] Huerta, A.; Fernández-Méndez, S.: Coupling Element Free Galerkin and Finite Element Methods. *ECCOMAS 2000*, CIMNE, Barcelona, 11.-14. September 2000.
- [65] Huerta, A.; Fernández-Méndez, S.: Enrichment and Coupling of the Finite Element and Meshless Methods. *Internat. J. Numer. Methods Engrg.*, **48**, 1615 – 1636, 2000.
- [66] Huerta, A.; Fernández-Méndez, S.; Liu, W.K.: A comparison of two formulations to blend finite elements and mesh-free methods. *Comp. Methods Appl. Mech. Engrg.*, **193**, 1105 – 1117, 2004.
- [67] Huerta, A.; Vidal, Y.; Villon, P.: Pseudo-divergence-free element free Galerkin method for incompressible fluid flow. *Comp. Methods Appl. Mech. Engrg.*, **193**, 1119 – 1136, 2004.
- [68] Idelsohn, S.R.; Oñate, E.; Calvo, N.; Del Pin, F.: Meshless finite element ideas. *Proceedings of the Fifth World Congress on Computational Mechanics (WCCM V)*, Vienna, 2002.

- [69] Idelsohn, S.R.; Oñate, E.; Calvo, N.; Pin, F. Del: The meshless finite element method. *Internat. J. Numer. Methods Engrg.*, **58**, 893 – 912, 2003.
- [70] Jiang, B.N.: *The least-squares finite element method—theory and applications in computational fluid dynamics and electromagnetics*. Springer Verlag, Berlin, 1998.
- [71] Jin, X.; Li, G.; Aluru, N.R.: Positivity conditions in meshless collocation methods. *Comp. Methods Appl. Mech. Engrg.*, **193**, 1171 – 1202, 2004.
- [72] Johnson, G.R.; Stryk, R.A.; Beissel, S.R.: SPH for high velocity impact computations. *Comp. Methods Appl. Mech. Engrg.*, **139**, 347 – 373, 1996.
- [73] Kaljevic, I.; Saigal, S.: An improved element free Galerkin formulation. *Internat. J. Numer. Methods Engrg.*, **40**, 2953 – 2974, 1997.
- [74] Klaas, O.; Shepard, M.S.: An Octree Based Partition of Unity Method for Three Dimensional Problems. *Fifth U.S. National Congress on Computational Mechanics*, Boulder, CO, 1999.
- [75] Krongauz, Y.; Belytschko, T.: Enforcement of Essential Boundary Conditions in Meshless Approximations Using Finite Elements. *Comp. Methods Appl. Mech. Engrg.*, **131**, 133 – 145, 1996.
- [76] Krongauz, Y.; Belytschko, T.: A Petrov-Galerkin Diffuse Element Method (PG DEM) and its comparison to EFG. *Comput. Mech.*, **19**, 327 – 333, 1997.
- [77] Krongauz, Y.; Belytschko, T.: EFG approximations with discontinuous derivatives. *Internat. J. Numer. Methods Engrg.*, **41**, 1215 – 1233, 1998.
- [78] Krysl, P.; Belytschko, T.: Element-free Galerkin method: Convergence of the continuous and discontinuous shape functions. *Comp. Methods Appl. Mech. Engrg.*, **148**, 257 – 277, 1997.
- [79] Krysl, P.; Belytschko, T.: An Efficient Linear-precision Partition of Unity Basis for Unstructured Meshless Methods. *Commun. Numer. Meth. Engrg.*, **16**, 239 – 255, 2000.
- [80] Krysl, P.; Belytschko, T.: ESFLIB: A Library to Compute the Element Free Galerkin Shape Functions. *Comp. Methods Appl. Mech. Engrg.*, **190**, 2181 – 2205, 2001.

- [81] Kulasegaram, S.; Bonet, J.; Lok, T.-S.L.; Rodriguez-Paz, M.: Corrected Smooth Particle Hydrodynamics - A Meshless Method for Computational Mechanics. *ECCOMAS 2000*, CIMNE, Barcelona, 11.-14. September 2000.
- [82] Lancaster, P.; Salkauskas, K.: Surfaces Generated by Moving Least Squares Methods. *Math. Comput.*, **37**, 141 – 158, 1981.
- [83] Leem, K.H.; Oliveira, S.; Stewart, D.E.: Some Numerical Results from Meshless Linear Systems. Technical report, Univeristy of Iowa, 2001.
- [84] Levin, D.: The approximation power of moving least-squares. *Math. Comput.*, **67**, 1517 – 1531, 1998.
- [85] Li, S.; Liu, W.K.: Moving least-squares reproducing kernel method Part II: Fourier analysis. *Comp. Methods Appl. Mech. Engrg.*, **139**, 159 – 193, 1996.
- [86] Li, S.; Liu, W.K.: Reproducing Kernel Hierarchical Partition of Unity, Part I – Formulation and Theory. *Internat. J. Numer. Methods Engrg.*, **45**, 251 – 288, 1999.
- [87] Li, S.; Liu, W.K.: Meshfree and particle methods and their applications. *Appl. Mech. Rev.*, **55**, 1 – 34, 2002.
- [88] Li, S.; Lu, H.; Han, W.; Liu, W.K.; Simkins, D.C.: Reproducing kernel element method. Part II: Globally conforming I^m/C^n hierarchies. *Comp. Methods Appl. Mech. Engrg.*, **193**, 953 – 987, 2004.
- [89] Li, X.-Y.; S.-H.Teng; Üngör, A.: Point Placement for Meshless Methods using Sphere Packing and Advancing Front Methods. *ICCES'00*, Los Angeles, USA, 20.-25. August 2000.
- [90] Li, X.-Y.; S.-H.Teng; Üngör, A.: Biting: Advancing Front Meets Sphere Packing. *Internat. J. Numer. Methods Engrg.*, **49**, 61 – 81, 2000.
- [91] Liu, G.R.: *Meshless Methods*. CRC Press, Boca Raton, 2002.
- [92] Liu, G.R.; Gu, T.: Meshless local Petrov-Galerkin (MLPG) method in combination with finite element and boundary element approaches. *Comput. Mech.*, **26**, 536 – 546, 2000.
- [93] Liu, W.K.; Belytschko, T.; Oden, J.T. (eds.): Meshless methods. *Comp. Methods Appl. Mech. Engrg.*, **139**(1-4, special issue), 1 – 400, 1996.

- [94] Liu, W.K.; Chen, Y.: Wavelet and Multiple Scale Reproducing Kernel Methods. *Int. J. Numer. Methods Fluids*, **21**, 901 – 931, 1995.
- [95] Liu, W.K.; Chen, Y.; Uras, R.A.; Chang, C.T.: Generalized Multiple Scale Reproducing Kernel Particle Methods. *Comp. Methods Appl. Mech. Engrg.*, **139**, 91 – 157, 1996.
- [96] Liu, W.K.; Han, W.; Lu, H.; Li, S.; Cao, J.: Reproducing kernel element method. Part I: Theoretical formulation. *Comp. Methods Appl. Mech. Engrg.*, **193**, 933 – 951, 2004.
- [97] Liu, W.K.; Jun, S.; Li, S.; Adee, J.; Belytschko, T.: Reproducing Kernel Particle Methods for Structural Dynamics. *Internat. J. Numer. Methods Engrg.*, **38**, 1655 – 1679, 1995.
- [98] Liu, W.K.; Jun, S.; Zhang, Y.F.: Reproducing Kernel Particle Methods. *Int. J. Numer. Methods Fluids*, **20**, 1081 – 1106, 1995.
- [99] Liu, W.K.; Li, S.; Belytschko, T.: Moving Least Square Reproducing Kernel Methods (I) Methodology and Convergence. *Comp. Methods Appl. Mech. Engrg.*, **143**, 113 – 154, 1997.
- [100] Liu, W.K.; Uras, R.A.; Chen, Y.: Enrichment of the finite element method with the reproducing kernel particle method. *J. Appl. Mech., ASME*, **64**, 861 – 870, 1997.
- [101] Lu, H.; Chen, J.S.: Adaptive Galerkin particle method. In *Meshfree Methods for Partial Differential Equations*. (Griebel, M.; Schweitzer, M.A., Eds.), Vol. 26, Springer Verlag, Berlin, 2002.
- [102] Lu, H.; Li, S.; Simkins, D.C.; Liu, W.K.; Cao, J.: Reproducing kernel element method. Part III: Generalized enrichment and application. *Comp. Methods Appl. Mech. Engrg.*, **193**, 989 – 1011, 2004.
- [103] Lu, Y.Y.; Belytschko, T.; Gu, L.: A New Implementation of the Element Free Galerkin Method. *Comp. Methods Appl. Mech. Engrg.*, **113**, 397 – 414, 1994.
- [104] Lucy, L.B.: A numerical approach to the testing of the fission thesis. *Astronom. J.*, **82**(12), 1013 – 1024, 1977.
- [105] Luo, Y.; Häussler-Combe, U.: An adaptivity procedure based on the gradient of strain energy density and its application in meshless methods. In *Meshfree Methods for Partial Differential Equations*. (Griebel, M.; Schweitzer, M.A., Eds.), Vol. 26, Springer Verlag, Berlin, 2002.

- [106] Melenk, J.M.; Babuška, I.: The Partition of Unity Finite Element Method: Basic Theory and Applications. *Comp. Methods Appl. Mech. Engrg.*, **139**, 289 – 314, 1996.
- [107] Monaghan, J.J.: Why particle methods work. *SIAM J. Sci. Comput.*, **3**, 422 – 433, 1982.
- [108] Monaghan, J.J.: An introduction to SPH. *Comput. Phys. Comm.*, **48**, 89 – 96, 1988.
- [109] Mukherjee, Y.X.; Mukherjee, S.: On Boundary Conditions in the Element-free Galerkin Method. *Comput. Mech.*, **19**, 264 – 270, 1997.
- [110] Muravin, B.; Turkel, E.: Advance diffraction method as a tool for solution of complex non-convex boundary problems. In *Meshfree Methods for Partial Differential Equations*. (Griebel, M.; Schweitzer, M.A., Eds.), Vol. 26, Springer Verlag, Berlin, 2002.
- [111] Nayroles, B.; Touzot, G.; Villon, P.: Generalizing the Finite Element Method: Diffuse Approximation and Diffuse Elements. *Comput. Mech.*, **10**, 307 – 318, 1992.
- [112] Noguchi, H.; Kawashima, T.; Miyamura, T.: Element free analyses of shell and spatial structures. *Internat. J. Numer. Methods Engrg.*, **47**, 1215 – 1240, 2000.
- [113] Oden, J.T.; Duarte, C.A.; Zienkiewicz, O.C.: A New Cloud-based hp Finite Element Method. *Comp. Methods Appl. Mech. Engrg.*, **153**, 117 – 126, 1998.
- [114] Oñate, E.; Idelsohn, S.; Zienkiewicz, O.C.; Taylor, R.L.: A Finite Point Method in Computational Mechanics. Applications to Convective Transport and Fluid Flow. *Internat. J. Numer. Methods Engrg.*, **39**, 3839 – 3866, 1996.
- [115] Oñate, E.; Idelsohn, S.; Zienkiewicz, O.C.; Taylor, R.L.; Sacco, C.: A Stabilized Finite Point Method for Analysis of Fluid Mechanics Problems. *Comp. Methods Appl. Mech. Engrg.*, **139**, 315 – 346, 1996.
- [116] Organ, D.; Fleming, M.; Terry, T.; Belytschko, T.: Continous meshless approximations for nonconvex bodies by diffraction and transparency. *Comput. Mech.*, **18**, 225 – 235, 1996.

- [117] Park, S.H.; Youn, S.K.: The least-squares meshfree method. *Internat. J. Numer. Methods Engrg.*, **52**, 997 – 1012, 2001.
- [118] Rabczuk, T.; Belytschko, T.; Xiao, S.P.: Stable particle methods based on Lagrangian kernels. *Comp. Methods Appl. Mech. Engrg.*, **193**, 1035 – 1063, 2004.
- [119] Randles, P.W.; Libersky, L.D.: Smoothed Particle Hydrodynamics: Some recent improvements and applications. *Comp. Methods Appl. Mech. Engrg.*, **139**, 375 – 408, 1996.
- [120] Sibson, R.: A vector identity for the Dirichlet tessellation. *Math Proc Cambridge Philos Soc*, **87**, 151 – 155, 1980.
- [121] Simkins, D.C.; Li, S.; Lu, H.; Liu, W.K.: Reproducing kernel element method. Part IV: Globally compatible C^n ($n \geq 1$) triangular hierarchy. *Comp. Methods Appl. Mech. Engrg.*, **193**, 1013 – 1034, 2004.
- [122] Sladek, V.; Sladek, J.; Atluri, S.N.; Keer, R. Van: Numerical Integration of Singularities in Meshless Implementation of Local Boundary Integral Equations. *Comput. Mech.*, **25**, 394 – 403, 2000.
- [123] Strouboulis, T.; Babuška, I.; Copps, K.: The design and analysis of the Generalized Finite Element Method. *Comp. Methods Appl. Mech. Engrg.*, **181**, 43 – 69, 2000.
- [124] Strouboulis, T.; Copps, K.; Babuška, I.: The Generalized Finite Element Method. *Comp. Methods Appl. Mech. Engrg.*, **190**, 4081 – 4193, 2001.
- [125] Sukumar, N.; Moran, B.; Belytschko, T.: The natural element method in solid mechanics. *Internat. J. Numer. Methods Engrg.*, **43**(5), 839 – 887, 1998.
- [126] Sukumar, N.; Moran, B.; Semenov, A.Y.; Belikov, V.V.: Natural neighbour Galerkin methods. *Internat. J. Numer. Methods Engrg.*, **50**, 1 – 27, 2001.
- [127] Swegle, J.W.: Smoothed Particle Hydrodynamics Stability Analysis. *J. Comput. Phys.*, **116**, 123 – 134, 1995.
- [128] Wagner, G.J.; Liu, W.K.: Application of essential boundary conditions in mesh-free methods: A corrected collocation method. *Internat. J. Numer. Methods Engrg.*, **47**, 1367 – 1379, 2000.

- [129] Wagner, G.J.; Liu, W.K.: Hierarchical enrichment for bridging scales and mesh-free boundary conditions. *Internat. J. Numer. Methods Engrg.*, **50**, 507 – 524, 2001.
- [130] Wu, C.K.C.; Plesha, M.E.: Essential boundary condition enforcement in meshless methods: Boundary flux collocation method. *Internat. J. Numer. Methods Engrg.*, **53**, 499 – 514, 2002.
- [131] Zhang, X.; Liu, X.H.; Song, K.Z.; Lu, M.W.: Least-squares collocation meshless method. *Internat. J. Numer. Methods Engrg.*, **51**, 1089 – 1100, 2001.
- [132] Zhu, T.; Zhang, J.; Atluri, S.N.: A Meshless Local Boundary Integral Equation (LBIE) Method for Solving Nonlinear Problems. *Comput. Mech.*, **22**, 174 – 186, 1998.
- [133] Zhu, T.; Zhang, J.-D.; Atluri, S.N.: A Local Boundary Integral Equation (LBIE) Method in Computational Mechanics, and a Meshless Discretization Approach. *Comput. Mech.*, **21**, 223 – 235, 1998.

1998-07	J. Schönwälder, M. Bolz, S. Mertens, J. Quittek, A. Kind, J. Nicklisch	SMX - Script MIB Extensibility Protocol Version 1.0
1998-08	C. Heimann, S. Lauterbach, T. Förster	Entwurf und Implementierung eines verteilten Ansatzes zur Lösung langrechnender Optimierungsprobleme aus dem Bereich der Ingenieurwissenschaften
1999-01	A. Zeller	Yesterday, my program worked. Today, it does not. Why?
1999-02	P. Niebert	A Temporal Logic for the Specification and Verification of Distributed Behaviour
1999-03	S. Eckstein, K. Neumann	Konzeptioneller Entwurf mit der Unified Modeling Language
1999-04	T. Gehrke, A. Rensink	A Mobile Calculus with Data
2000-01	T. Kaiser, B. Fischer, W. Struckmann	The Modula-2 Proving System MOPS
2000-02	J. Saperia, J. Schönwälder	Policy-Based Enhancements to the SNMP Framework
2000-03	A. Casties	Finite-Element-Interpolation der räumlichen Dichten eines Vielteilchensystems auf ungeordneten Gittern
2000-04	J. Koslowski	A 2-dimensional view of the Chu-construction
2000-05	S. Eckstein, P. Ahlbrecht, K. Neumann	Von parametrisierten Spezifikationen zu generierten Informationssystemen: ein Anwendungsbeispiel
2000-06	F. Strauß, J. Schönwälder, M. Mertens	JAX - A Java AgentX Sub-Agent Toolkit
2000-07	F. Strauß	Advantages and Disadvantages of the Script MIB Infrastructure
2000-08	T. Gehrke, U. Goltz	High-Level Sequence Charts with Data Manipulation
2000-09	T. Firley	Regular languages as states for an abstract automaton
2001-01	K. Diethers	Tool-Based Analysis of Timed Sequence Diagrams
2002-01	R. van Glabbeek, U. Goltz	Well-behaved Flow Event Structures for Parallel Composition and Action Refinement
2002-02	J. Weimar	Translations of Cellular Automata for Efficient Simulation
2002-03	H. G. Matthies, M. Meyer	Nonlinear Galerkin Methods for the Model Reduction of Nonlinear Dynamical Systems
2002-04	H. G. Matthies, J. Steindorf	Partitioned Strong Coupling Algorithms for Fluid-Structure-Interaction
2002-05	H. G. Matthies, J. Steindorf	Partitioned but Strongly Coupled Iteration Schemes for Nonlinear Fluid-Structure Interaction
2002-06	H. G. Matthies, J. Steindorf	Strong Coupling Methods
2002-07	H. Firley, U. Goltz	Property Preserving Abstraction for Software Verification
2003-01	M. Meyer, H. G. Matthies	Efficient Model Reduction in Non-linear Dynamics Using the Karhunen-Loève Expansion and Dual-Weighted-Residual Methods
2003-02	C. Täubner	Modellierung des Ethylen-Pathways mit UML-Statecharts
2003-03	T.-P. Fries, H. G. Matthies	Classification and Overview of Meshfree Methods